# UNDERSTANDING HUMAN MOTION: RECOGNITION AND RETRIEVAL OF HUMAN ACTIVITIES

A DISSERTATION SUBMITTED TO

THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Nazlı İkizler

May, 2008

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assist. Prof. Dr. Pınar Duygulu(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Özgür Ulusoy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assoc. Prof. Dr. Aydın Alatan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Enis Çetin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. H. Altay Güvenir

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

# ABSTRACT

## UNDERSTANDING HUMAN MOTION: RECOGNITION AND RETRIEVAL OF HUMAN ACTIVITIES

Nazlı İkizler
Ph.D. in Computer Engineering
Supervisor: Assist. Prof. Dr. Pınar Duygulu
May, 2008

Within the ever-growing video archives is a vast amount of interesting information regarding human action/activities. In this thesis, we approach the problem of extracting this information and understanding human motion from a computer vision perspective. We propose solutions for two distinct scenarios, ordered from simple to complex. In the first scenario, we deal with the problem of single action recognition in relatively simple settings. We believe that human pose encapsulates many useful clues for recognizing the ongoing action, and we can represent this shape information for 2D single actions in very compact forms, before going into details of complex modeling. We show that high-accuracy single human action recognition is possible 1) using spatial oriented histograms of rectangular regions when the silhouette is extractable, 2) using the distribution of boundary-fitted lines when the silhouette information is missing. We demonstrate that, inside videos, we can further improve recognition accuracy by means of adding local and global motion information. We also show that within a discriminative framework, shape information is quite useful even in the case of human action recognition in still images.

Our second scenario involves recognition and retrieval of complex human activities within more complicated settings, like the presence of changing background and viewpoints. We describe a method of representing human activities in 3D that allows a collection of motions to be queried without examples, using a simple and effective query language. Our approach is based on units of activity at segments of the body, that can be composed across time and across the body to produce complex queries. The presence of search units is inferred automatically by tracking the body, lifting the tracks to 3D and comparing to models trained using motion capture data. Our models of short time scale limb behaviour are built using labelled motion capture set. Our

query language makes use of finite state automata and requires simple text encoding and no visual examples. We show results for a large range of queries applied to a collection of complex motion and activity. We compare with discriminative methods applied to tracker data; our method offers significantly improved performance. We show experimental evidence that our method is robust to view direction and is unaffected by some important changes of clothing.

*Keywords:* Human motion, action recognition, activity recognition, activity retrieval, image and video processing, classification.

# ÖZET

# İNSAN HAREKETİNİ ANLAMA: İNSAN AKTİVİTELERİNİN TANINMASI VE ERİŞİMİ

Nazlı İkizler
Bilgisayar Mühendisliği, Doktora
Tez Yöneticisi: Yrd. Doç. Dr. Pınar Duygulu
Mayıs, 2008

Sürekli olarak büyüyen video arşivlerinde insan hareketleri ve aktiviteleriyle ilgili çok geniş miktarda ilginç bilgi bulunmaktadır. Bu tezde, bu bilgileri elde etme ve insan hareketini anlama konusuna bilgisayarlı görü açısından yaklaşıyoruz. Bu amaçla, kolaydan zora doğru sıralanan iki ayrı senaryo için çözümler öneriyoruz. İlk senaryoda, nispeten kolay sayılabilecek durumlardaki teksel aksiyon tanıma problemini ele almaktayız. Bu senaryo için, insan duruşunun varolan aktiviteyi tanımlamak için pekçok faydalı ipucu içerdigine inanıyoruz ve iki boyutlu aksiyonlar için karmaşık modellemeye gitmeden, bu şekil bilgisini çok kompakt biçimlerde gösterebiliriz. Bu kapsamda, yüksek doğruluk oranlı insan aksiyonu tanımanının mümkün olduğunu 1) videolardan siluet bilgisi çıkarmanın mümkün olduğu durumlarda dikdörtgensel alanların uzamsal yönelimli histogramlarını kullanarak, 2) siluet bilgisi bulunmadığı durumlarda sınırlardan çıkarılmış çizgilerin dağılımlarını kullanarak gösteriyoruz. Buna ek olarak, videolarda, tanıma doğruluğunu yerel ve genel hareket bilgisi eklemek suretiyle geliştirebileceğimizi kanıtlıyoruz. Şekil bilgisinin ayrıştırıcı bir çerçeve dahilinde, durağan resimlerdeki insan hareketlerini tanıma probleminde bile oldukça faydalı olduğunu gösteriyoruz.

İkinci senaryo karmaşık insan aktivitelerinin, değişen arka plan ve görüş açıları gibi komplike durumlarda tanınması ve erişimi konularını içermektedir. Böyle durumlarda üç boyutlu insan aktiviteleri betimlemek ve bir hareket derlemesini görsel örneğe ihtiyaç olmaksızın sorgulamak için bir yöntem tanımlıyoruz. Yaklaşımımız, vücut bölümleri üzerinde oluşturulan ve zamansal ve uzamsal olarak düzenlenebilecek aktivite birimlerine dayanmaktadır. Arama birimlerinin varlığı, önce insan vücudunun takibi, bu takip izlerinin üçüncü boyuta taşınması ve hareket algılama verisi üzerinde öğrenilmiş modellerle karşılaştırmak yolu ile otomatik olarak sağlanmaktadır. Kısa zamanlı uzuv davranış modellerimiz etiketlenmiş hareket algılama veri kümesi kullanılarak oluşturulmaktadır. Video sorgu dilimiz sonlu durumlu özdevinirlerden

faydalanmaktadır ve sadece basit metin kodlamasıyla tanımlanabilir olup görsel örneğe ihtiyaç duymamaktadır. Çalışmamızda karmaşık hareket ve aktivite derlemesine uyguladığımız geniş aralıktaki sorguların sonuçlarını sunuyoruz. Kendi yöntemimizi izleme verisi üzerine uygulanmış ayrıştırıcı yöntemlerle karşılaştırıyoruz; ve yöntemimizin belirgin derecede gelişmiş performans sergilediğini gösteriyoruz. Deneysel kanıtlarımız, yöntemimizin görüş yönü farklılıklarına dayanıklı olduğunu ve kıyafetlerdeki önemli değişikliklerinden etkilenmediğini ispatlamaktadır.

*Anahtar sözcükler*: İnsan hareketi, aksiyon tanıma, aktivite tanıma, aktivite erişimi, resim ve video işleme, sınıflandırma.

# Acknowledgement

This was a journey. Actually, the initial part of a longer one. Took longer than estimated, tougher than expected. But, you know what they say: “No pain, no gain”. I learnt a lot, and it was all worth it.

During this journey, my pathway crossed with lots of wonderful people. The very first one is Dr. Pınar Duygulu-Şahin, who has been a great advisor for me. Her passion for research, for computer vision and for living has been a true inspiration. She has provided tremendous opportunities for my research career and I am deeply thankful for her guidance, encouragement and motivation in each and every way.

I was one of the lucky people, who had the chance to meet and work with Prof. David A. Forsyth. Words cannot express my gratitude to him. I learnt a lot from his vast knowledge. He is exceptional, both as a scientist and as a person.

I am grateful to the members of my thesis committee, Prof. Aydın Alatan, Prof. Özgür Ulusoy, Prof Enis Çetin and Prof. H. Altay Güvenir for accepting to read and review this thesis and for their valuable comments. I am also thankful to Dr. Selim Aksoy, whose guidance has been helpful in many ways.

I would like to acknowledge the financial support of TÜBİTAK (Scientific and Technical Research Council of Turkey) during my visit in University of Illinois at Urbana-Champaign(UIUC) as a research scholar. This research has also been partially supported by TÜBİTAK Career grant number 104E065 and grant numbers 104E077 and 105E065.

I am deeply thankful to Deva Ramanan, for sharing his codes and invaluable technical knowledge. This thesis has benefitted a lot from the landmarks he set on tracking and pose estimation research. Neither my research nor my days in University of Illinois at Urbana-Champaign would be complete, without the presence and endless support of dear Shadi, Alex and Nicolas. I cannot thank them enough for their friendship, their motivation and support.

I am also thankful to the exquisite members of RETINA research group. Selen,

Fırat, Aslı and Daniya made room EA522 feel like home. Their enthusiasm was a great motivation. I am also grateful my other friends, especially Emre and Tağmaç, for their understanding and encouragement.

Above all, I owe everything to my parents. None of this would be possible, without their unconditional love and endless support. My mother (Aysun) and my father (Aykut), thank you for nurturing me with love, with the curiosity for learning and research, for being my inspiration in every dimension of life, and for giving me the strength to carry on during the hard times of this journey. I am blessed to be your daughter. I am also blessed by the presence of my brother(Nuri) in my life. Thank you for all the laughter and joy.

And finally, my reserved thanks are to Gökberk, for all the good and the harmony.

*To my parents,*
*Aysun and Aykut İkizler*

# Contents

**1 Introduction** **1**

1.1 Organization of the Thesis . . . . . 7

**2 Background and Motivation** **9**

2.1 Application Areas . . . . . 11

2.2 Human Motion Understanding in Videos . . . . . 12

2.2.1 Timescale . . . . . 12

2.2.2 Motion primitives . . . . . 13

2.2.3 Methods with explicit dynamical methods . . . . . 14

2.2.4 Methods with partial dynamical models . . . . . 15

2.2.5 Discriminative methods . . . . . 15

2.2.6 Transfer Learning . . . . . 17

2.2.7 Histogramming . . . . . 17

2.3 Human Action Recognition in Still Images . . . . . 17

2.3.1 Pose estimation . . . . . 18

2.3.2 Inferring actions from poses . . . . . 19

**3 Recognizing Single Actions 20**

3.1 Histogram of Oriented Rectangles as a New Pose Descriptor . . . . . 21

3.1.1 Extraction of Rectangular Regions . . . . . 22

3.1.2 Describing Pose as Histograms of Oriented Rectangles . . . . 23

3.1.3 Capturing Local Dynamics . . . . . 24

3.1.4 Recognizing Actions with HORs . . . . . 25

3.2 The Absence of Silhouettes: Line and Flow Histograms for Human Action Recognition . . . . . 30

3.2.1 Line-based shape features . . . . . 32

3.2.2 Motion features . . . . . 35

3.2.3 Recognizing Actions . . . . . 36

3.3 Single Action Recognition inside Still Images . . . . . 37

3.3.1 Pose extraction from still images . . . . . 38

3.3.2 Representing the pose . . . . . 41

3.3.3 Recognizing Actions in Still Images . . . . . 42

**4 Experiments on Single Human Actions 43**

4.1 Datasets . . . . . 43

4.1.1 Video Datasets . . . . . 44

4.1.2 Still Image Datasets . . . . . 45

4.2 Experiments with Histogram of Oriented Rectangles (HORs) . . . . . 49

4.2.1 Optimal Configuration of the Pose Descriptor . . . . . 49

4.2.2 Classification Results and Discussions . . . . . 53

4.2.3 Comparison to other methods and HOGs . . . . . 54

4.2.4 Computational Evaluation . . . . . 58

4.3 Experiments with Line and Flow Histograms . . . . . 58

4.4 Experiments on Still Images . . . . . 61

**5 Recognizing Complex Human Activities 67**

5.1 Representing Acts, Actions and Activities . . . . . 69

5.1.1 Acts in short timescales . . . . . 70

5.1.2 Limb action models . . . . . 70

5.1.3 Limb activity models . . . . . 71

5.2 Transducing the body . . . . . 73

5.2.1 Tracking . . . . . 73

5.2.2 Lifting 2D tracks to 3D . . . . . 74

5.2.3 Representing the body . . . . . 76

5.3 Querying for Activities . . . . . 81

**6 Experiments on Complex Human Activities 87**

6.1 Experimental Setup . . . . . 87

6.1.1 Datasets . . . . . 88

6.1.2 Evaluation Method . . . . . 90

6.2 Expressiveness of Limb Activity Models . . . . . 91

6.2.1 Vector Quantization for Action Dynamics . . . . . 93

6.3 Searching . . . . . 94

6.3.1 Torso exclusion . . . . . 95

6.3.2 Controls . . . . . 97

6.4 Viewpoint evaluation . . . . . 98

6.5 Activity Retrieval with Complex Backgrounds . . . . . 103

**7 Conclusions and Discussion** **105**

7.1 Future Directions . . . . . 107

7.2 Relevant Publications . . . . . 109

# List of Figures


1.1 Example of a single action. . . . . 2

1.2 Example of a complex activity, composed across time and across the body. . . . . 4

2.1 Earliest work on human motion photography by Eadweard Muybridge [63, 64]. . . . . 10

2.2 Possible application areas of human action recognition . . . . . 10

3.1 Feature extraction stage of our histogram of rectangles(HOR) approach 22

3.2 Rectangle extraction step . . . . . 23

3.3 Details of histogram of oriented rectangles (HORs) . . . . . 24

3.4 Nearest neighbor classification process for a walking sequence. . . . 26

3.5 Global rectangle images formed by summing the whole sequence . . . 27

3.6 SVM classification process over a window of frames . . . . . 28

3.7 Dynamic Time Warping (DTW) over 2D histograms . . . . . 30

3.8 Two-level classification with mean horizontal velocity and SVMs (v+SVM) . . . . . 31

3.9 Extraction of line-based features . . . . . 33

3.10 Forming line histograms . . . . . 33

3.11 Forming optical flow(OF) histograms . . . . . 35

3.12 Overall system architecture with addition of mean horizontal velocity. 37

3.13 Actions in still images. . . . . 38

3.14 Pose and rectangle extraction. **To the left:** The original image and its corresponding parse obtained by using iterative parsing as defined in [74]. **To the right:** The extracted silhouette and the resulting rectangles. . . . . 39

3.15 Pose representation using circular histogram of oriented rectangles(CHORs). Circular grid is centered to the maximum value of the probability parse. . . . . 41

4.1 Example frames from the Weizzman dataset introduced in [12]. . . . 44

4.2 Example frames from the KTH dataset introduced in [86]. . . . . 46

4.3 Extracted silhouettes from the KTH dataset in s1 recording condition. 47

4.4 Example images of the ActionWeb dataset collected from the web sources. . . . . 48

4.5 Example frames from the figure skating dataset introduced in [100]. . 49

4.6 Rectangle detection with torso exclusion . . . . . 52

4.7 Confusion matrices for each matching method over the Weizzman dataset 55

4.8 Confusion matrix for classification results of the KTH dataset. . . . . 56

4.9 Choice of $\alpha$ and resulting confusion matrix for the KTH dataset. . . . 59

4.10 Resulting confusion matrix for the KTH dataset. . . . . 60

4.11 Examples for correctly classified images of actions running, walking, throwing, catching, crouching, kicking in consecutive lines. . . . . 63

4.12 Confusion matrix of CHOR method over the ActionWeb still images dataset . . . . . 64

4.13 Examples for misclassified images of actions for ActionWeb dataset . 65

4.14 Clusters formed by our approach for the figure skating dataset. . . . . 66

5.1 Overall system architecture for the retrieval of complex human activities. . . . . 68

5.2 Formation of activity models for each of the body parts. . . . . 72

5.3 Example good tracks for the UIUC video dataset. . . . . 73

5.4 Out-of-track examples in the UIUC dataset. . . . . 73

5.5 Posterior probability map of a `walk-pickup-carry` video of an arm. . . . . 76

5.6 An example query result of our system. . . . . 79

5.7 Another example sequence from our system, performed by a female subject. . . . . 80

5.8 The FSA for a single action is constructed based on its unit length. . . 82

5.9 Here, example query FSAs for a sequence where the subject walks into the view, stops and waves and then walks out of the view are shown. . 84

5.10 Query FSA for a video where the person walks, picks something up and carries it. . . . . 85

6.1 Example frames from UIUC complex activity dataset. . . . . 89

6.2 Example frames from our dataset of single activities with different viewspoints. . . . . 89

6.3 Example frames from the Friends dataset which is compiled from the Friends TV Series . . . . . 90

6.4 Average HMM posteriors for the motion capture dataset . . . . . 93

6.5 The choice of $k$ in k-means . . . . . 94

6.6 Effect of torso inclusion . . . . . 96

6.7 The results of ranking for 15 queries over our video collection. . . . . 99

6.8 Average precision values for each query. . . . . 100

6.9 Evaluation of the sensitivity to viewpoint change . . . . . 101

6.10 Mean precisions and the PR curves for each action . . . . . 102

6.11 Example tracks for the Friends dataset . . . . . 103

6.12 Results of our retrieval system over the Friends dataset. . . . . 104

# List of Tables


4.1 The accuracies of the matching methods with respect to angular bins (over a grid of $3 \times 3$ . . . . . 50

4.2 The accuracies of the matching methods with respect to $N \times N$ grids 51

4.3 Overall performance of the matching methods over the Weizzman and KTH datasets. . . . . 53

4.4 Comparison of our method to other methods that have reported results over the Weizzman dataset. . . . . 56

4.5 Comparison of our method to other methods that have reported results over KTH dataset. . . . . 57

4.6 Comparison to HOG feature based action classification over the KTH dataset. . . . . 58

4.7 Run time evaluations for different matching techniques using HORs. 58

4.8 Comparison of our method to other methods on KTH dataset. . . . . 60

4.9 Comparison with respect to recording condition of the videos in the KTH dataset. . . . . 61

6.1 Our collection of video sequences, named by the instructions given to actors. . . . . 88

6.2 The Mean Average Precison(MAP) values for different types of queries. . . . . 95

# Chapter 1

# Introduction

This thesis tries to address the problem of understanding what people are doing, which is one of the great unsolved problems of computer vision. A fair solution opens tremendous application possibilities, ranging from medical to security. The major difficulties have been that (a) good kinematic tracking is hard; (b) models typically have too many parameters to be learned directly from data; and (c) for much everyday behaviour, there isn't a taxonomy. This thesis aims to tackle with this problem in the prevalence of these difficulties, while presenting solutions to various cases.

We approach the problem of understanding human motion in two distinct scenarios, ordered simple to complex, with respect to difficulty level. While choosing these scenarios, we try to comply with the requirements of the ongoing research trends in the action recognition community and the real-world activities. With this intention, we first cover the case of recognizing single actions, where the person in video(or image) is involved in one simple (non-complex) action. Figure 1.1 illustrates an example occurance of a single "walking" action. Our second scenario involves recognition and retrieval of complex human activities within more complicated settings, like the presence of changing background and viewpoints. This scenario is more realistic than the simple one, and covers a large portion of the available video archives which involve full-body human activities.

We deal with the simpler scenario as our first area of interest, because the current

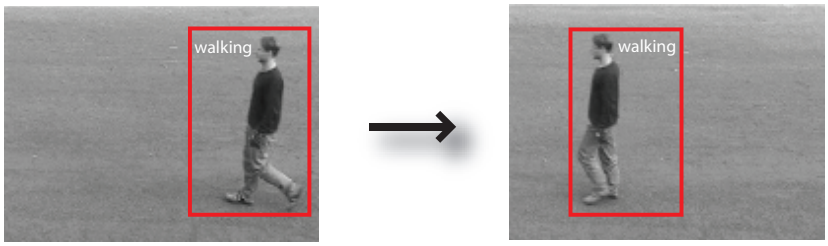



Figure 1.1: Example of a single action.

research in vision community has condensed around “one actor, one action, simple background” paradigm. This is mostly due to the lack of the benchmark datasets that cover the remaining aspects of this subject, and due to the extreme challenges of processing the complicated settings. This paradigm is by no means a representative of the available videos at hand, and only a small portion of the real world videos meet the requirements stated. However, we can say that it is a good starting point for observing the nature of human actions from a machine vision perspective.

There are three key elements that define a single action:

- pose of the body (and parts)
- relative ordering of the poses
- speed of the body (and parts)

We can formulate single action recognition as a mixture of these three elements. The relative importance of these elements is based on the nature of the actions that we aim to recognize. For example, if we want to differentiate an instance of a “bend” action from a “walk” action, the pose of the human figure gives sufficient information. However, if we want to discriminate between “jog” and “run” actions, the pose alone may not be enough, due to the similarity in the nature of these actions in the pose domain. In such cases, the speed information needs to be incorporated. Various attempts in action recognition literature try to model some or all of these aspects. For instance, methods based on spatio-temporal templates mostly pay attention to the pose of the human body, whereas methods based on dynamical models focus their attention to modeling the ordering of these poses in greater detail.

We believe that the human pose encapsulates many useful clues for recognizing the ongoing action. Even a single image may convey quite rich information for understanding the type of action taking place. Actions can mostly be represented by configurations of the body parts, before building complex models for understanding the dynamics.

Using this idea, we base the foundation of our method on defining the pose of the human body to discriminate single actions, and by introducing new pose descriptors, we try to evaluate how far we can go only with a good description of the pose of the body in 2D. We evaluate two distinct cases here: The presence of silhouette information in the domain, and the absence of silhouettes. We also evaluate how our system benefits from adding the remaining action components whenever necessary.

For the case where silhouette information is easily extractable, we use rectangular regions as our basis of shape descriptor. Unlike most of the methods that use complex modeling of body configurations, we follow the analogy of Forsyth *et al.* [32], which represents the body as a set of rectangles, and explore the layout of these rectangles. Our pose descriptor is based on a similar intuition: the human body can be represented by a collection of oriented rectangles in the spatial domain and the orientations of these rectangles form a signature for each action. However, rather than detecting and learning the exact configuration of body parts, we are only interested in the distribution of the rectangular regions which may be the candidates for the body parts.

When we cannot extract the silhouette information from the image sequences, due to various reasons like camera movement, zoom effect, etc., but the background is relatively simple and the boundaries are identifiable, we propose to use a compact shape representation based on boundary-fitted lines. We show how we can make use of our new shape descriptor together with a dense representation of optical flow and global temporal information for robust single action recognition. Our representation involves a very compact form by making use of feature reduction techniques, decreasing the classification time significantly.

Recognizing single actions is a relatively simpler problem compared to complex activities; it is relatively easier to acquire training data for identifying single actions. In addition, current datasets available only deal with static backgrounds where foreground

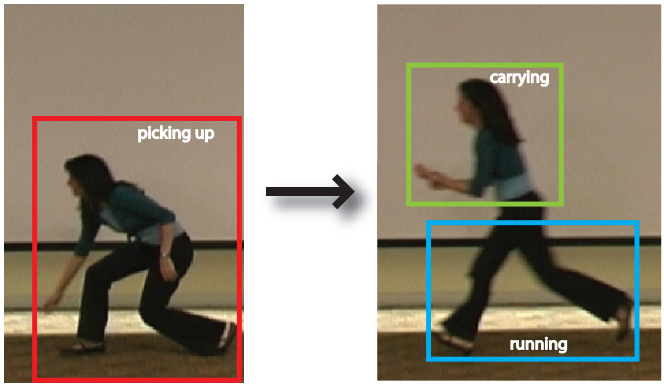



Figure 1.2: Example of a complex activity, composed across time and across the body.

human figures are easily extractable for further processing. Under these circumstances, we believe that a very compact representation should be enough to conform the needs of single action recognition, and presenting such compact representations is what we do in the first part of this thesis.

In the second part of the thesis, we consider the case of complex activity recognition, where the action units are composed over time and space and the viewpoints of the subjects are changing frequently. Figure 1.2 shows an example complex composite activity, in which the person performs two different activities consecutively and one activity is the composite of two different actions. Desirable properties of a complex activity recognition and retrieval system are:

- it should handle different clothings and varying motion speeds of different actors
- it should accomodate the different timescales over which actions are sustained
- it should allow composition across time and across the body
- there should be a manageable number of parameters to estimate
- it should perform well in presence of limited quantities of training data
- it should be indifferent to viewpoint changes
- it should require no example video segment for querying

Building such a system has many practical applications. For example, if a suspicious behaviour can be encoded in terms of "action word"s - w.r.t. arms and legs separately whenever needed - one can submit a text query and search for that specific behaviour within security video recordings. Similarly, one can encode medically critical behaviours and search for those in surveillance systems.

Understanding activities is a complex issue in many aspects. First of all, there is a shortage of training data, because a wide range of variations of behaviour is possible. A particular nuisance is the tendency of activity to be compositional (below). Discriminative methods on appearance may be confounded by intraclass variance. Different subjects may perform the actions with different speeds in various outfits and and these nuisance variations make it difficult to work directly with appearance. Training a generative model directly on a derived representation of video is also fraught with difficulty. Either one must use a model with very little expressive power (for example, an HMM with very few hidden states) or one must find an enormous set of training data to estimate dynamical parameters (the number of which typically goes as the square of the number of states). This issue has generated significant interest in variant dynamical models.

The second difficulty is the result of the composite nature of activities. Most of the current literature on activity recognition deals with simple actions. However, real life involves more than just simple "walk"s. Many activity labels can meaningfully be composed, both over time — "walk"ing then "run"ing — and over the body — "walk"ing while "wave"ing. The process of composition is not well understood (see the review of animation studies in [33]), but is a significant source of complexity in motion. Examples include: "walking while scratching head" or "running while carrying something". Because composition makes so many different actions possible, it is unreasonable to expect to possess an example of each activity. This means we should be able to find activities for which we do not possess examples.

A third issue is that tracker responses are noisy, especially when the background is cluttered. For this reason, discriminative classifiers over tracker responses work poorly. One can boost the performance of discriminative classifiers if they are trained on noise-free environments, like carefully edited motion capture datasets. However,

these will lack the element of compositionality.

All these points suggest having a model of activity which consists of **pieces** which are relatively easily learned and are then combined together within a model of **composition**. In this study, we try to achieve this by

- learning local dynamic models for atomic actions distinctly for each body part, over a motion capture dataset
- authoring a compositional model of these atomic actions
- using the emissions of the data with these composite models

To overcome the data shortage problem, we propose to make use of motion capture data. This data does not consist of everyday actions, but rather a limited set of American football movements. There is a form of transfer learning problem here — we want to learn a model in a football domain and apply it to an everyday domain — and we believe that transfer learning is an intrinsic part of activity understanding.

We first author a compositional model for each body part using a motion capture dataset. This authoring is done in a similar fashion to phoneme-word conjunctions in speech recognition: We join atomic action models to have more complex activity models. By doing so, we achieve the minimum of parameter estimation. In addition, composition across time and across body is achieved by building separate activity models for each body part. By providing composition across time and space, we can make use of the available data as much as possible and achieve a broader understanding about what the subject is up to.

After forming the compositional models over 3D data, we track the 2D video with a state-of-the-art full body tracker and lift 2D tracks to 3D, by matching the snippets of frames to motion capture data. We then infer activities with these lifted tracks. By this lifting procedure, we achieve view-invariance, since our body representation is in 3D.

Finally, we write text queries to retrieve videos. In this procedure, we do not require example videos and we can query for activities that have never been seen before.

Making use of finite state automata, we employ a simple and effective query language that allows complex queries to be written in order to retrieve the desired set of activity videos. Using separate models for each body part, compositional nature of our system allows us to span a huge query space.

Here, our particular interest is everyday activity. In this case, a fixed vocabulary either doesn't exist, or isn't appropriate. For example, one often does not know words for behaviours that appear familiar. One way to deal with this is to work with a notation (for example, laban notation); but such notations typically work in terms that are difficult to map to visual observables (for example, the weight of a motion). We must either develop a vocabulary or develop expressive tools for authoring models. We favour this third approach.

We compare our method with several controls, and each of these controls has a discriminative form. First, we built discriminative classifiers over raw 2D tracks. We expect that discriminative methods applied to 2D data perform poorly because intra-class variance overwhelms available training data. In comparison, our method benefits by being able to estimate dynamical models on motion capture dataset. Second, we built classifiers over 3D lifts. Although classifiers applied to 3D data could be view invariant, we expect poor performance because there is not much labelled data and the lifts are noisy. Our third control involves classifiers trained on 3D motion capture data and applied to lifted data. This control also performs poorly, because noise in the lifting process is not well represented by the training data. This also causes problems with the composition. On contrary, our model supports a high level of composition and its generative nature handles different lengths of actions easily. In the corresponding experiments chapter, we evaluate the effect of all these issues and also analyze the view-invariance of our method in greater detail.

## 1.1 Organization of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 starts with a brief introduction to human action/activity recognition research together with possible application areas. It includes an overview of human action/activity recognition approaches in the literature.

Chapter 3 describes our approaches to recognition of single human actions within relatively simple scenarios. By single actions, we mean the videos including only one action instance. Particularly, Section 3.1 and Section 3.2 introduce our histogram-based approaches for single action recognition in videos, whereas Section 3.3 includes application of our pose descriptor to still images. In Chapter 4, we present our methods' performance on single action recognition case.

Later on, Chapter 5 introduces our approaches for understanding human actions in the case of complex scenarios. These scenarios include actions composed across body and across space, with varying viewpoints and cluttered backgrounds. We show how we can handle those scenarios within a 3D modeling approach. Chapter 6 gathers up our empirical evaluations of our method on complex human activities.

Chapter 7 concludes the thesis with a summary and discussions of the approaches presented and delineates possible future directions.

# Chapter 2

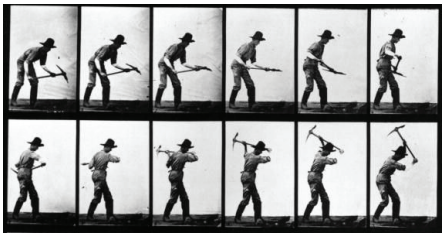
# Background and Motivation

Immense developments in video technology, both recording (as in TiVo and surveillance systems) and broadcasting (as in YouTube [1]), have greatly increased the size of accessable video archives, thus, the demand on processing and extracting useful information from those archives. Although the demand is quite high, the relevant searches still depends on text-based user annotations, and visual properties mostly go untouched. While using annotations is a sensible approach, not all the videos are annotated, or existing annotations/metadata are useful.

Inside those video archives is a vast amount of interesting information regarding human action/activities.

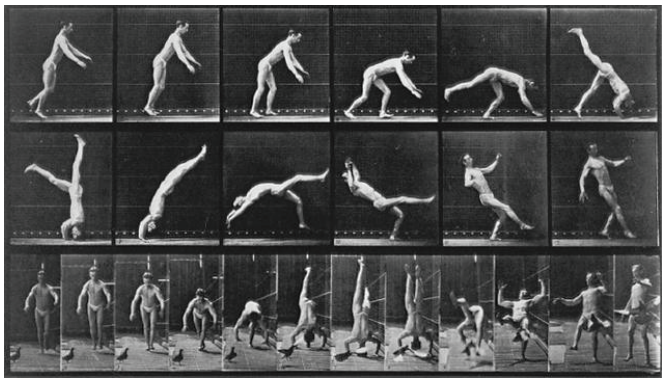
From a pscyhological perspective, the presence of human figure inside images is quite important. We can observe this importance from the extensive literature and history of face recognition research(see [85, 110]). People are interested in identification and recognition of humans and their actions. Starting from the works of Eadweard Muybridge [63], as early as 1887 (Figure 2.1), movement and action analysis and synthesis has captured a lot of interest, which resulted in the development of motion pictures.

Additionally, understanding what people are doing will close the semantic gap between low-level features and high-level image interpretation a great extent.



(a) pickaxe man

(b) dancing woman

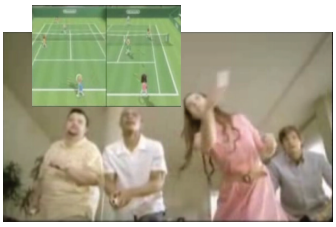


(c) head-spring

Figure 2.1: Earliest work on human motion photography by Eadweard Muybridge [63, 64].

All these make automatic understanding of human motion a very important problem for computer vision research. In the rest of the chapter, we present a summary of the related studies over this subject.

(a) surveillance



(b) home entertainment


(c) sport annotation

Figure 2.2: Possible application areas of human action recognition

## 2.1 Application Areas

Human motion understanding can serve many application areas, ranging from visual surveillance to human computer interaction(HCI) systems. Particularly, the application domains are limited to those that involve camera setups. Below is an example list of such systems.

- Visual Surveillance: As the video technology become more commonplace, visual surveillance systems undertook a rapid development process, and have more or less become a part of our daily lives. Figure 2.2(a) shows an example surveillance infra-red (IR) video output. Human action understanding can help to find fraudful events –such as burglaries, fightings, etc –, to detect pedestrains from moving vehicles, and can serve to track patients who need special attention (like detecting a falling person [94]).

- Human-Computer Interaction: Ubiquitous computing has increased the presence of HCI systems everywhere. A recently evolving thread is in the area of electronic games and home entertainment(see Figure 2.2(b)). These systems are currently based on very naive video and signal processing. However, as the technology evolve, the trend will shift towards more intelligent and sophisticated HCI systems which involve activity and behaviour understanding.

- Sign Language Recognition: Gesture recognition, which is a subdomain of action recognition that operates over the upper body parts, serves a lot for automatic understanding of sign language [15, 38, 92].

- News, Movie and Personal Video Archives: By the decrease in the cost of video capturing devices and by the development of sharing websites, videos become to be a substantial part of the today's personal visual archives. Automatic annotations of those archives, together with movie and news archives will help information retrieval. In addition, automatic annotation of news and sport video archives(see Figure 2.2(c) for an example frame) is a necessary thread for accessing the necessary information in a quick and easy way. People may be interested in finding certain events, describable only by the activities involved and activity recognition can help considerably in this case.

- Social Evaluation of Movements: The observation of behavioural patterns of humans is quite important for the research of sociology, architecture, and more. Machine perception of activities and patterns can guide many researches in this area. For example, Yan *et al.* tries to find estimates of where people spend time by examining head trajectories [105]. Interestingly, research like this one will help in urban planning.

## 2.2 Human Motion Understanding in Videos

There is a long tradition of research on interpreting human actions and activities in the computer vision community. Especially during the last decade, human action recognition has gained a lot of interest. Hu *et al* [43] and Forsyth *et al* [33] present extensive surveys on this subject.

In general, approaches to human action and activity recognition on videos can be divided intro three main threads. First, one can use motion primitives(Section 2.2.2) which is based on the statistical evaluation of motion clusters. Second, one can use dynamical models, partially(Section 2.2.4) or explicitly(Section 2.2.3). Third, one can make use of discriminative methods(Section 2.2.5), such as spatio-temporal templates or "bag-of-words". This section presents a literature overview of these methods.

### 2.2.1 Timescale

Regarding the timescale of the act, action and activity descriptions, there is a wide range of helpful distinctions. Bobick [13] distinguishes between movements, activity and actions, corresponding to longer timescales and increasing complexity of representation; some variants are described in two useful review papers [4, 36]. In this thesis, we refer short-timescale representations as **acts**, like a forward-step or a hand-raise; medium timescale movements as **actions**, like walking, running, jumping, standing,

waving, and long timescale movements as **activities**. Activities are complex composites of actions, whereas actions are typically composites of multiple acts. The composition can be across time (sequential ordering of acts/actions) and across body(body parts involving in different acts/actions).

### 2.2.2 Motion primitives

A natural method for building models of motion on longer time scales is to identify clusters of motion of the same type and then consider the statistics of how these *motion primitives* are strung together. There are pragmatic advantages to this approach: we may need to estimate fewer parameters and can pool examples to do so; we can model and account for long term temporal structure in motion; and matching may be easier and more accurate. Feng and Perona describe a method that first matches motor primitives at short timescales, then identifies the activity by temporal relations between primitives [30]. In animation, the idea dates at least to the work of Rose *et al.*, who describe motion *verbs* — our primitives — and *adverbs* — parameters that can be supplied to choose a particular instance from a scattered data interpolate [82]. Primitives are sometimes called *movemes*. Matarić *et al.* represent motor primitives with force fields used to drive controllers for joint torque on a rigid-body model of the upper body [59, 60]. Del Vecchio *et al.* define primitives by considering all possible motions generated by a parametric family of linear time-invariant systems [98]. Barbič *et al.* compare three motion segmenters, each using a purely kinematic representation of motion [9]. Their method moves along a sequence of frames adding frames to the pool, computing a representation of the pool using the first $k$ principal components, and looking for sharp increases in the residual error of this representation. Fod *et al.* construct primitives by segmenting motions at points of low total velocity, then subjecting the segments to principal component analysis and clustering [31]. Jenkins and Mataric segment motions using kinematic considerations, then use a variant of Isomap (detailed in [48]) that incorporates temporal information by reducing distances between frames that have similar temporal neighbours to obtain an embedding for kinematic variables [47]. Li *et al.* segment and model motion capture data simultaneously using a linear dynamical system model of each separate primitive and a Markov model to

string the primitives together by specifying the likelihood of encountering a primitive given the previous primitive [56].

### 2.2.3 Methods with explicit dynamical methods

Hidden Markov Models (HMM's) have been very widely adopted in activity recognition, but the models used have tended to be small (e.g, three and five state models in [19]). Such models have been used to recognize: tennis strokes [104]; pushes [101]; and handwriting gestures [106]. Toreyin *et al.* [94] use HMMs for falling person detection, by fusing audial and visual information together. Feng and Perona [30] call actions "movelets", and build a vocabulary by vector quantizing a representation of image shape. These codewords are then strung together by an HMM, representing activities; there is one HMM per activity, and discrimination is by maximum likelihood. The method is not view invariant, depending on an image centered representation. There has been a great deal of interest in models obtained by modifying the HMM structure, to improve the expressive power of the model without complicating the processes of learning or inference. Methods include: coupled HMM's ([19]; to classify T'ai Chi moves); layered HMM's ([67]; to represent office activity); hierachies ([62]; to recognize everyday gesture); HMM's with a global free parameter ([102]; to model gestures); and entropic HMM's ([18]; for video puppetry). Building variant HMM's is a way to simplify learning the state transition process from data (if the state space is large, the number of parameters is a problem). But there is an alternative — one could author the state transition process in such a way that it has relatively few free parameters, despite a very large state space, and then learn those parameters; this is the lifeblood of the speech community.

Stochastic grammars have been applied to find hand gestures and location tracks as composites of primitives [17]. However, difficulties with tracking mean that there is currently no method that can exploit the potential view-invariance of lifted tracks, or can search for models of activity that compose across the body and across time.

Finite state methods have been used directly. Hongeng *et al.* demonstrate recognition of multi-person activities from video of people at coarse scales (few kinematic

details are available); activities include conversing and blocking [40]. Zhao and Nevatia use a finite-state model of walking, running and standing, built from motion capture [109]. Hong *et al.* use finite state machines to model gesture [38].

### 2.2.4 Methods with partial dynamical models

Pinhanez and Bobick [69, 70] describe a method for detecting activities using a representation derived from Allen's interval algebra [5], a method for representing temporal relations between a set of intervals. One determines whether an event is past, now or future by solving a consistent labelling problem, allowing temporal propagation. There is no dynamical model — sets of intervals produced by processes with quite different dynamics could be a consistent labelling; this can be an advantage at the behaviour level, but probably is a source of difficulties at the action/activity level. Siskind [89] describes methods to infer activities related to objects — such as throw, pick up, carry, and so on — from an event logic formulated around a set of physical primitives —- such as translation, support relations, contact relations, and the like — from a representation of video. A combination of spatial and temporal criteria are required to infer both relations and events, using a form of logical inference. Shi *et al.* make use of Propagation Nets to encode the partial temporal orderings of actions [88]. Recently, Ryoo and Aggarwal use context-free grammars to exploit the temporal relationships between atomic actions to define composite activities [84].

### 2.2.5 Discriminative methods

Methods with (partial/explicit) dynamical models mostly have a generative nature. This section outlines the approaches which have a discriminative setting. These methods mostly rely on 2D local image features.

#### 2.2.5.1 Methods based on Templates

The notion that a motion produces a characteristic spatio-temporal pattern dates at least to Polana and Nelson [71]. Spatio-temporal patterns are used to recognize actions in [14]. Ben-Arie *et al.* [10] recognize actions by first finding and tracking body parts using a form of template matcher and voting on lifted tracks. Bobick and Wilson [16] use a state-based method that encodes gestures as a string of vector-quantized observation segments; this preserves order, but drops dynamical information. Efros *et al.* [26] use a motion descriptor based on optical flow of a spatio-temporal volume, but their evaluation is limited to matching videos only. Blank *et al.* [12] define actions as space-time volumes. An important disadvantage of methods that match video templates directly is that one needs to have a template of the desired action to perform a search. Ye *et al.* moves one step further in this aspect and use matching by parts, instead of using the whole volumetric template [50]. However, their part detection is manual.

#### 2.2.5.2 Bag-of-words approaches

Recently, 'bag-of-words' approaches originated from text retrieval research is being adopted to action recognition. These studies are mostly based on the idea of forming codebooks of 'spatio-temporal' features. Laptev *et al.* first introduced the notion of 'space-time interest points' [53] and used SVMs to recognize actions [86]. P. Dollár *et al.* extract cuboids via separable linear filters and form histograms of these cuboids to perform action recognition [25]. Niebles *et al.* applied a pLSA approach over these patches (i.e. the cuboids extracted with the method of [25]) to perform unsupervised action recognition [66]. Recently, Wong *et al.* proposed using pLSA method with and implicit shape model to infer actions from spatio-temporal codebooks [103]. They also show the superior performance of applying SVMs for action recognition. However, these methods are not viewpoint independent and very likely to suffer from complex background schemes.

### 2.2.6 Transfer Learning

Recently, transfer learning has become a very hot research topic in machine learning community. It is based on transfering the information learned from one domain to the another related domain. In one of the earlier works, Caruana approached this problem by discovering common knowledge shared between tasks via "multi-task learning" [20]. Evgeniou and Pontil [27] utilize SVMs for multi-task learning. Ando and Zhang [6] generate some artificial auxiliary tasks to use shared prediction structures between similar tasks. A recent application involves transfering American Sign Language(ASL) words learned from a synthetic dictionary to real world data [28].

### 2.2.7 Histogramming

Histogramming is an old trick that has been frequently used in computer vision research. For action recognition, Freeman and Roth [35] use orientation histograms for hand gesture recognition. Recently, Dalal and Triggs use histograms of oriented gradients (HOGs) for human detection in images [22], which is shown to be quite successful. Later on, Dalal *et al.* make use of HOGs together with orientation histograms of optical flow for human detection in videos [23]. Christian Thurau [93] evaluate HOGs within a motion primitive framework and use histograms of HOGs as the representation of the videos for action recognition. Zu *et al.*, on the other hand, utilizes histograms of optical flow in forms of slices to recognize actions in tennis videos [111]. Recently, Dedeoğlu *et al.* define a silhouette-based shape descriptor and use histogram of the matched poses for action recognition [24].

## 2.3 Human Action Recognition in Still Images

Most of the effort on understanding the human actions involves video analysis with fundamental applications such as surveillance and human computer interaction. However, action recognition on single images is a mostly ignored area. This is due to various challenges of this topic. The lack of region model in a single image precludes

discrimination of foreground and background objects. The presence of articulation makes the problem much harder, for there is a large number of alternatives for the human body configuration. Humans as being articulated objects, can exhibit various poses, resulting in high variability of the images. Thus, the problem of action recognition on still images becomes a very challenging problem.

### 2.3.1 Pose estimation

Recognition of actions from still images starts with finding the person within the image and inferring the pose of it. There are many studies in finding person images( [46]), localizing the persons ([3]), or pedestrian detection([95]). Dalal and Triggs propose a very successful edge and gradient based descriptor, called Histogram of Oriented Gradients [22] for detecting and locating humans in still images. Zhu *et al.* advances HOG descriptors by integrating HOG and AdaBoost to select the most suitable block for detection [112]. In [11], Bissacco *et al.* also use HOGs in combination with Latent Dirichlet Allocation for human detection and pose estimation. Oncel *et al.* [96], on the other hand, define a covariance descriptor for human detection.

For inferring the human pose from 2D images, there is a bunch of recent studies. Most of the studies are dealing with cases where human figure is easily differentiable from the background, i.e. using a non-cluttered stable background. Those studies include inferring 3D pose from 2D image data, as in [2] where Agarwal *et al.* deal with inferring 3D pose from silhouettes. Rosales *et al.* estimate the 3D pose from a silhouette using multi-view data [81]. In [87], a method based on hashing for finding relevant poses in a database of images is presented.

Over the domain of cluttered images, Forsyth and Fleck introduce the concept of body plans as a representation for people and animals in complex environments [32]. Body plans view people and animals as assemblies of cylindrical parts. To learn such articulated body plans, [80] introduces using Support Vector Machines(SVMs) and Relevant Vector Machines(RVMs). Ramanan presents an iterative parsing process for pose estimation of articulated objects [74], which we use for extracting human parses from still images for action recognition. We discuss this method in greater detail in

Section 3.3.1.

Ren *et al.* also presents a framework for detecting and recovering human body configuration [79]. In their recent work, Zhang *et al.* describe a hierarchical model based on edge and skin/hair color features and deterministic and stochastic search [108].

### 2.3.2 Inferring actions from poses

To our best knowledge, there are quite few studies that deal with the problem of human action recognition in static images. Wang *et al.* partially addresses this problem [100]. They represent the overall shape as a collection of edges obtained through canny edge detection and propose a deformable matching method to measure distance of a pair of images. However, they only tackle the problem in an unsupervised manner and within single sports scenes.

# Chapter 3

# Recognizing Single Actions

This chapter presents the methods we developed for the recognition of single actions. By single actions, we refer to the action sequences where the human in motion is engaged with one action only, through the whole sequence. This chapter investigates this simpler case, and defines new pose descriptors which are very compact and easy to process. We define two shape-based features for this purpose. First one is applicable to the case where the silhouette information is easily extractable from the given sequence. The second pose descriptor handles the case when the silhouette information is not available in the scene.

We show how we can use these pose descriptors with various supervised and unsupervised approaches for action classification. In addition to video domain, we apply our pose descriptors for recognition of actions inside static images. Our main goal is to have compact, yet effective representations of single actions without going into complex modelling of dynamics. In the consecutive chapter, we show that our descriptors perform considerably well in the case of single action recognition with experimenting over various state-of-art datasets.

## 3.1 Histogram of Oriented Rectangles as a New Pose Descriptor

Following the body plan analogy of Forsyth *et al.* [32], which considers the body of the humans or animals as a collection of cylindrical parts, we represent the human body as a collection of rectangular patches and we base our motion understanding approach on the fact that the orientations and positions of these rectangles change over time with respect to the actions carried out. With this intuition, our algorithm first extracts rectangular patches over the human figure available in each frame, and then forms a spatial histogram of these rectangles by grouping over orientations. We then evaluate the changes of these histograms over time.

More specifically, given the video, first, the tracker identifies the location of the subject. Any kind of tracker, which can extract silhouette information of the humans can be used at this step. Using the extracted silhouettes, we search for the rectangular patches that can be candidates for the limbs. We do not discriminate between legs and arms here. Then, we divide the bounding box around the silhouette into an equal-sized grid and compute the histograms of the oriented rectangles inside each region. This bounding box is divided into $N \times N$ equal-sized spatial (grid) bins. While forming these spatial bins, the ratio between the body parts, i.e. head, torso and legs, is taken into account. At each time $t$, a pose is represented with a histogram $H_t$ based on the orientations of the rectangles in each spatial bin. We form our feature vector by combining the histograms from each subregion. This process is depicted in Fig. 3.1.

In the ideal case, single rectangles that fit perfectly to the limb areas should give enough information about the pose of the body. However, finding those perfect rectangles is not straightforward and is very prone to noise. Therefore, in order to eliminate the effect of noise, we use distribution of candidate rectangular regions as our feature. This gives a more precise information about the most probable locations of the fittest rectangles.

Having formed the spatio-temporal rectangle histograms for each video, we match any newly seen sequence to the examples at hand and label the videos accordingly. We

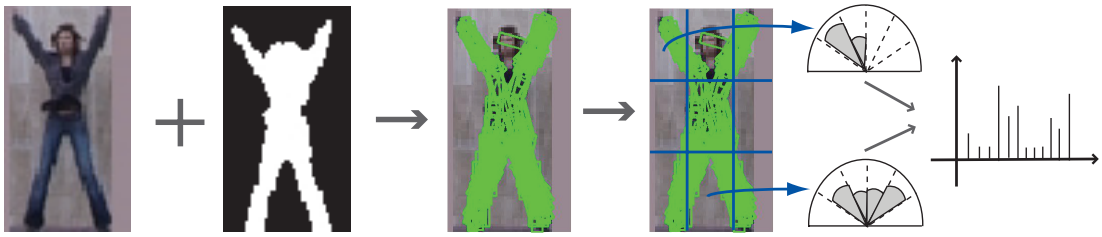


Figure 3.1: Here, the feature extraction stage of our approach is shown (this figure is best viewed in color). Using the extracted silhouettes, we search for the rectangular patches that can be candidates for the limb and compute the histograms of the oriented rectangles.

now describe the steps of our method in greater detail.

### 3.1.1 Extraction of Rectangular Regions

For describing the human pose, we make use of rectangular patches. These patches are extracted in the following way:

1) The tracker fires a response for the human figure and differentiates the human region from the background. This is usually done using a foreground-background discrimination method [34]. The simplest approach is to apply background subtraction, after forming a dependable model of the background. The reader is referred to [33] for a detailed overview of the subject. In our experiments, where we extract the silhouettes, we use a background subtraction scheme to localize the subject in motion, as in [37]. Note that any other method that extracts the silhouette of the subject will work just fine.

2) We then search for rectangular regions over the human silhouette using convolution of a rectangular filter on different orientations and scales. We make use of undirected rectangular filters, following Ramanan *et al.* [76]. The search is performed using 12 tilting angles, which are 15° apart, covering a search space of 180°. Note that since we don't have the directional information of these rectangle patches, orientations do not cover 360°, but its half. To tolerate the differences in the limb sizes and in the varying camera distances to the subject, we perform the rectangle convolution over multiple scales.

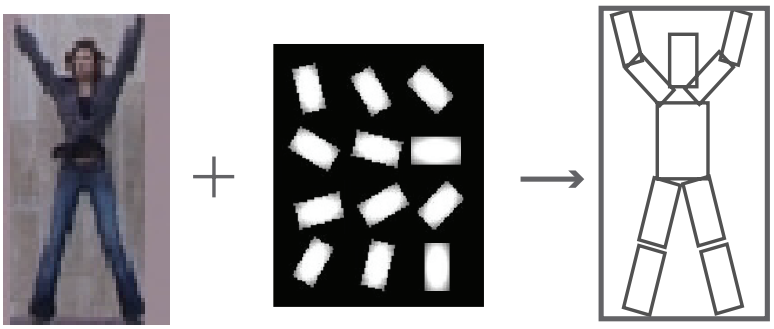


Figure 3.2: The rectangular extraction process is shown. We use zero-padded Gaussian filters with $15^\circ$ tilted orientations over the human silhouette. We search over various scales, without discriminating between different body parts. The perfect rectangular search for the given human subject would result in the tree structure to the right.

More formally, we form a zero-padded rectangular Gaussian filter $G_{rect}$ and produce the rectangular regions $R(x, y)$ by means of the convolution of the binary silhouette image $I(x, y)$ with this rectangle filter $G_{rect}$.

$$R(x, y) = G_{rect}(x, y) \circ I(x, y) \tag{3.1}$$

where $G_{rect}$ is a zero-padded rectangular patch of a 2-D Gaussian $G(x, y)$

$$G(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left(\frac{x^2}{{\sigma_x}^2} + \frac{y^2}{{\sigma_y}^2} - \frac{2\rho xy}{(\sigma_x\sigma_y)}\right)\right) \tag{3.2}$$

Higher response areas to this filter are more likely to include patches of a particular kind. The filters used are shown in Fig. 3.2.

To tolerate noise and imperfect silhouette extraction, this rectangle search allows a portion of the candidate regions to remain non-responsive to the filters. Regions that have low overall responses are eliminated this way. We then select the $k$ of the remaining candidate regions of each scale by random sampling (we used $k = 300$).

### 3.1.2 Describing Pose as Histograms of Oriented Rectangles

After finding the rectangular regions of the human body, in order to define the pose, we propose a simple pose descriptor, which is the Histogram of Oriented Rectangles

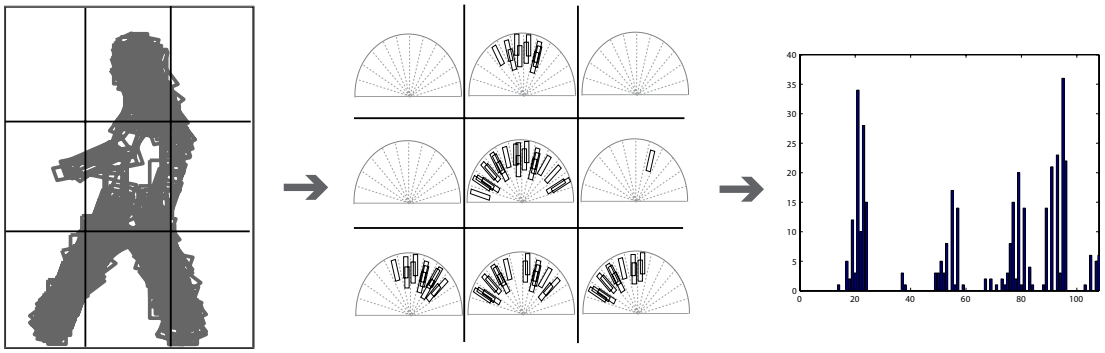


Figure 3.3: Details of histogram of oriented rectangles (HORs). The bounding box around the human figure is divided into an $N \times N$ grid (in this case, $3 \times 3$) and the HORs from each spatial bin are shown. The resulting feature vector is a concatenation of the HORs from each spatial bin.

(HOR). We compute the histogram of extracted rectangular patches based on their orientations. The rectangles are histogrammed over $15^\circ$ orientations, resulting in 12 circular bins. In order to incorporate spatial information of the human body, we evaluate these circular histograms within a $N \times N$ grid placed over the whole body. Our experiments show that $N = 3$ gives the best results. We form this grid by splitting the silhouette over the y-dimension based on the length of the legs. The area covering the silhouette is divided into equal-sized bins from bottom to up and left to right (see Fig. 3.3 for details). Note that, in this way, we give some space to the top part of the head, to allow action space for the arms (for actions like reaching, waving, etc.).

We have also evaluated the effects of using $30^\circ$ orientation bins and a $2 \times 2$ grid, which have more concise feature representations, but coarser detail of the human pose. We show the corresponding results in Sect. 4.2.

### 3.1.3 Capturing Local Dynamics

In action recognition, there may be times where one cannot discriminate two actions by just looking at single poses. In such cases, an action descriptor based purely on shape is not enough and temporal dynamics must be explored. To incorporate temporal features, HORs can be calculated over snippets of frames rather than single frames. More formally, we define histograms of oriented rectangles over a window of frames

(HORW), such that the histogram of the $i$th frame will be

$$HORW(i) = \sum_{k=i-n}^{i} HOR(k) \tag{3.3}$$

where $n$ is the size of the window.

By using HORs over a window of frames like this, we capture local dynamics information. In our experiments, we observe that, using HORWs is more useful especially to discriminate actions like "jogging" and "running", which are very similar in pose domain, but different in speed. Therefore, over a fixed length window, the compactness of these two actions will be different. We evaluate the effect of using HORs vs HORWs in greater detail in Section 4.2.

### 3.1.4 Recognizing Actions with HORs

After calculating the pose descriptors for each frame, we perform action classification in a supervised manner. There are four matching methods we perform in order to evaluate the performance of our pose descriptor in action classification problems.

#### 3.1.4.1 Nearest Neighbor Classification

The simplest scheme we utilize is to perform matching based on single frames (or snippets of frames in the case of HORWs), ignoring thedynamics of the sequence. That is, for each test instance frame, we find the closest frame in the training set and assign its label as the label of the test frame. We then employ a voting scheme throughout the whole sequence. This process is shown in Fig. 3.4. The pose descriptor of each frame(snippet) is compared to that of the training set frames and the closest frame's class is assigned as a label to that frame. The resulting is a vote vector, where each frame contributes with a vote and the majority class of the votes is the recognized action label for that sequence.

The distance between frames is computed using Chi-square distance between the histograms (as in [55]). Each frame with the histogram $H_i$ is labeled with the class of

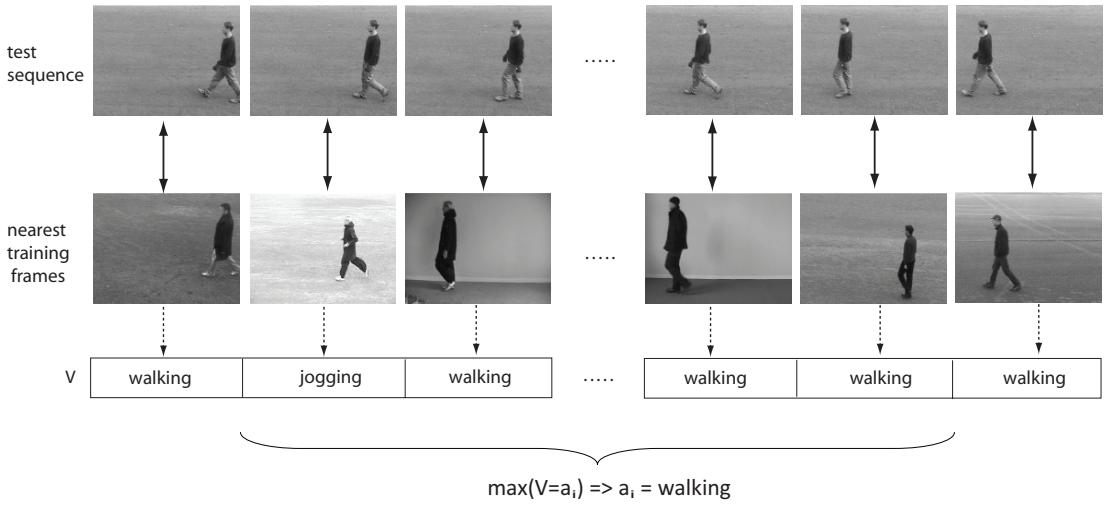



Figure 3.4: Nearest neighbor classification process for a walking sequence.

the frame having histogram $H_j$ that has the smallest distance $\chi^2$ such that

$$\chi^2(H_i, H_j) = \frac{1}{2}\sum_n \frac{(H_i(n) - H_j(n))^2}{H_i(n) + H_j(n)} \tag{3.4}$$

We should note that both $\chi^2$ and $L_2$ distance functions are very prone to noise, because a slight shift of the bounding box center of the human silhouette may result in a different binning of the rectangles and, therefore, may cause large fluctuations in distance. One can utilize Earth Mover's Distance [83] or Diffusion Distance [57], which are shown to be more efficient for histogram comparison in the presence of such shifts, by taking the distances between bins into account at the expense of higher computation time.

#### 3.1.4.2 Global Histogramming

Global histogramming is similar to the Motion Energy Image (MEI) method proposed by Bobick and Davis [14]. In this method, we sum up all spatial histograms of oriented rectangles through the sequence, and form a single compact representation for the entire video. This is simply done by collapsing all time information into a single dimension by summing the histograms and forming a global histogram $H_{global}$ such

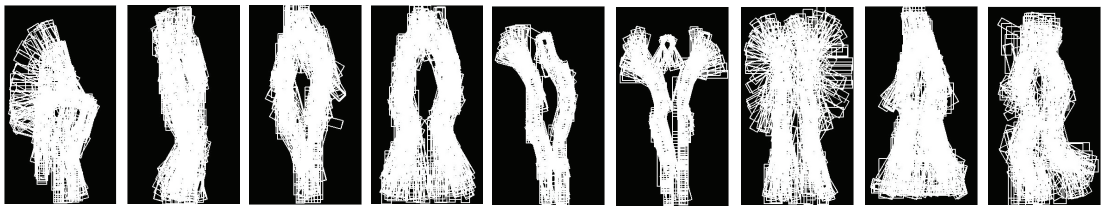


Figure 3.5: Global histograms are generated by summing up all the sequence and forming the spatial histograms of oriented rectangles from these global images. In this figure, global images after the extraction of the rectangular patches are shown for 9 separate action classes. These are bend, jump, jump in place, gallop sideways, one-hand wave, two-hands wave, jumpjack, walk and run actions.

that

$$H_{global}(d) = \sum_{t} H(d,t) \tag{3.5}$$

for each dimension $d$ of the histogram. Each test instance's $H_{global}$ is compared to that of the training instances using $\chi^2$ distance, and the label of the closest match is reported. The corresponding global images are shown in Fig. 3.5. These images show that for each action (of the Weizzman dataset in this case), even a simple representation like global histogramming can provide useful interpretations. These images resemble the Motion Energy Images of [14], however we do not use these shapes. Instead, we form the global spatial histogram of the oriented rectangles as our feature vector.

#### 3.1.4.3 Discriminative Classification - SVMs

Nearest neighbor schemes may fail to respond well to the complex classification problems. For this reason, we decided to make use of discriminative classification techniques. We pick Support Vector Machine(SVM) [97] classifiers from the pool of discriminative classifiers one could use, due to their reputation of success in various applications. We trained separate SVM classifiers for each action. These SVM classifiers are formed using *RBF* kernels over snippets of frames using a windowing approach. This process is depicted in Fig. 3.6. For choosing the parameters of the SVMs, we performa a grid search over the parameter space of the SVM classifiers and select the best classifiers using 10-fold cross validation. In our windowing approach, we segment the sequence into $k$-length chunks with some overlapping ratio $o$, and then classify these

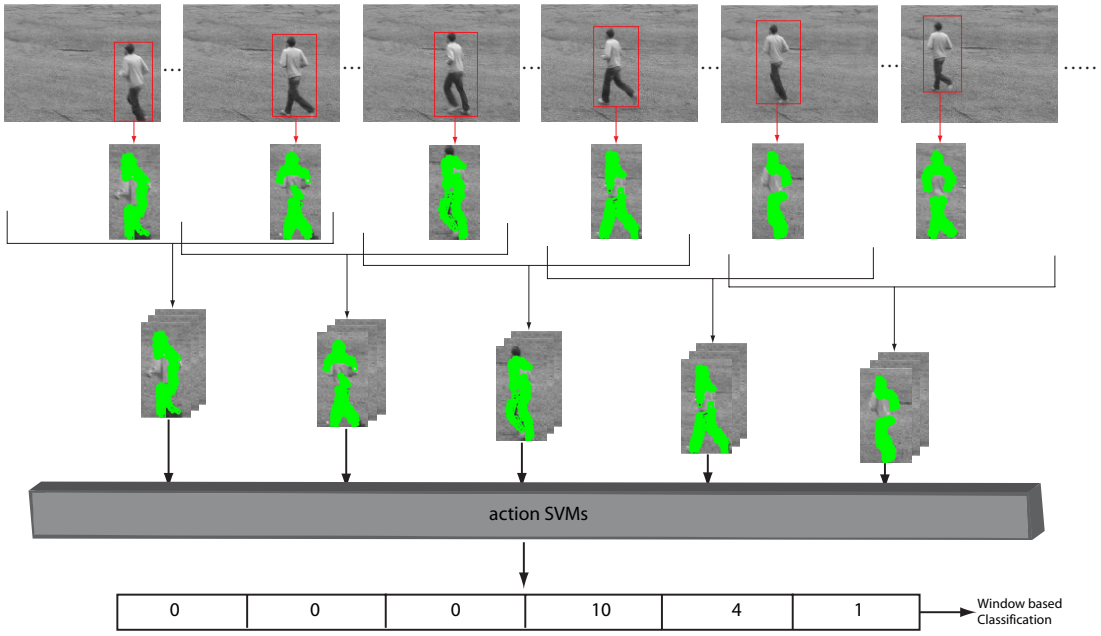



Figure 3.6: SVM classification process over a window of frames

chunks separately (we achieved the best results with $k = 15$ , and $o = 3$). The whole sequence is then labeled with the most frequent action class among its chunks.

#### 3.1.4.4 Dynamic Time Warping

Since the periods of the actions are not uniform, comparing sequences is not straightforward. In the case of human actions, the same action can be performed at different speeds, resulting in the sequence to be expanded or shrunk in time. In order to eliminate such effects of different speeds and to perform robust comparison, the sequences need to be aligned.

Dynamic time warping (DTW) is a method to compare two time series which may be different in length. DTW operates by trying to find the optimal alignment between two time series by means of dynamic programming (for more details, see [72]). The time axes are warped in such a way that samples of the corresponding points are aligned.

More specifically, given two time series $X = \{x_1 \dots x_n\}$ and $Y = \{y_1 \dots y_m\}$, the

distance $D(i,j)$ is calculated with

$$D(i,j) = \left\{ \begin{array}{l} D(i,j-1) \\ D(i-1,j) \\ D(i-1,j-1) \end{array} \right\} + d(x_i, y_j) \tag{3.6}$$

where $d(.,.)$ is the local distance function specific to application. In our implementation, we have chosen $d(.,.)$ as the $\chi^2$ distance function, as in Equation 3.4.

We use dynamic time warping along each dimension of the histograms separately. As shown in Fig. 3.7, we take each 1-d series of the histogram bins of the test video $X$ and compute the DTW distance $D(X(d), Y(d))$ to the corresponding 1-d series of the training instance $Y$. We try to align these sequences along each histogram dimension by DTW and report the sum of the smallest distances. Note that, separate alignment of each histogram bin also allows us to handle the fluctuations in distinct body part speeds. We then sum up the distances of all dimensions to compute the global DTW distance ($D_{global}$) between the videos. We label the test video with the label of the training instance that has the smallest $D_{global}$ such that,

$$D_{global}(X,Y) = \sum_{d=1}^{M} D(X(d), Y(d)) \tag{3.7}$$

where $M$ is the total number of bins in the histograms. While doing this, we exclude the top $k$ of the distances to reduce the effect of noise introduced by shifted bins and inaccurate rectangle regions. We choose $k$ based on the size of the feature vector such that $k = \lfloor \#num\_bins/2 \rfloor$ where $\#num\_bins$ is the total number of bins of the spatial grid.

#### 3.1.4.5 Classification with Global Velocity

When shape information is not enough, we can use speed information as a prior for action classes. Suppose we want to discriminate two actions: “handwaving” versus “running”. If the velocity of the person in motion is equal to zero, the probability that

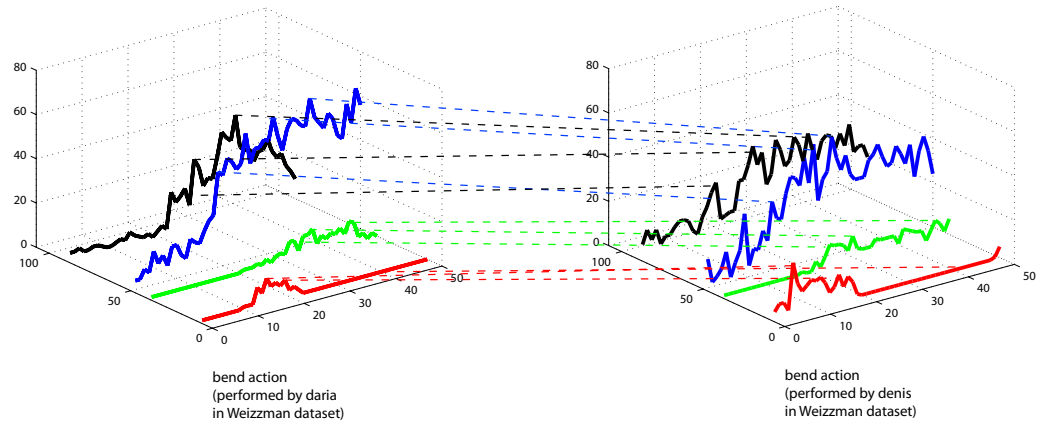



Figure 3.7: Dynamic Time Warping (DTW) over 2D histograms: We compute DTW distances between the histograms by evaluating the DTW cost over single dimensions separately and summing up all costs to get a global distance between sequences. Here, histograms of two bend actions performed by different actors are shown.

he has been running is quite low.

Based on this observation, we propose a two-level classification system. In the first level, we calculate mean velocities of the training sequences and fit a gaussian to each action in action set $A = \{a_1..a_n\}$ . Later on, given a test instance, we compute the posterior probability of each action $a_i \in A$ over these gaussians, and if the posterior probability of $a_i$ is greater than a threshold $t$(we use a loose bound $t = 0.1$), then we add $a_i$ to the probable set $S$ of actions for that sequence. After this preprocessing step, as the second level, we evaluate only the outputs of the SVMs for actions $a_k \in S$, and we take the maximum response from this subset of SVM classifiers as our classification decision. This process is shown in Fig. 3.8.

## 3.2 The Absence of Silhouettes: Line and Flow Histograms for Human Action Recognition

In the absence of silhouettes, we can make use of simpler features: lines. In this section, we present a pose descriptor based on the orientation of lines extracted from human boundaries. By using these lines together with optical flow information, we show

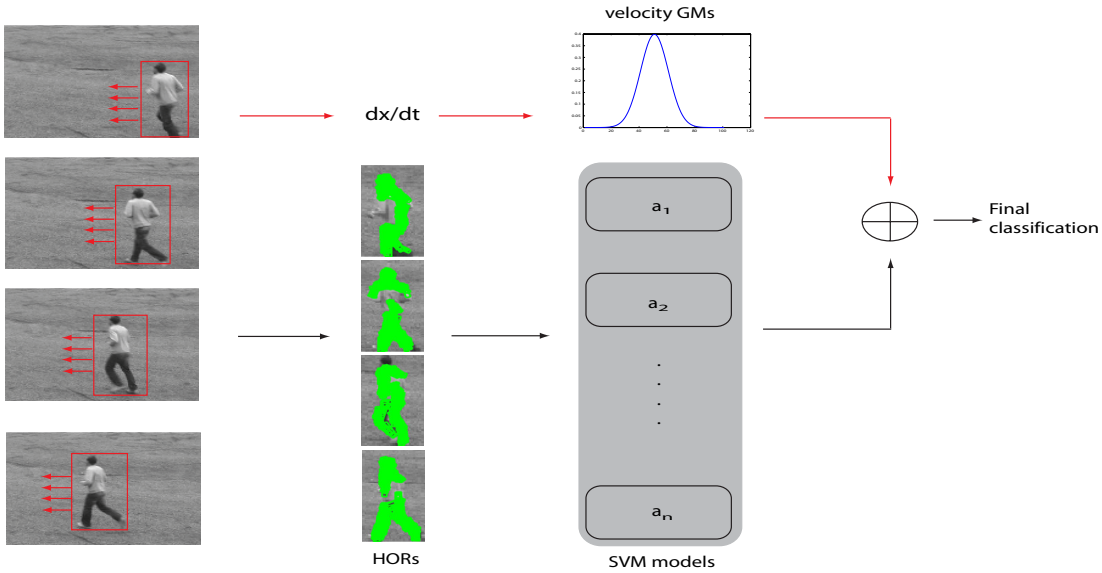



Figure 3.8: Two-level classification of actions based on mean horizontal velocity and histograms of oriented rectangles. First, the velocity of the subject is calculated throughout the entire video. We evaluate the posterior probability of this velocity and determine the probable set of actions for that video. Then, based on this probable set of actions, we look at the responses from corresponding SVM classifiers and take the maximum response as the classification label of that video.

that we can have fast and reliable action recognition, even if we don't have silhouette information.

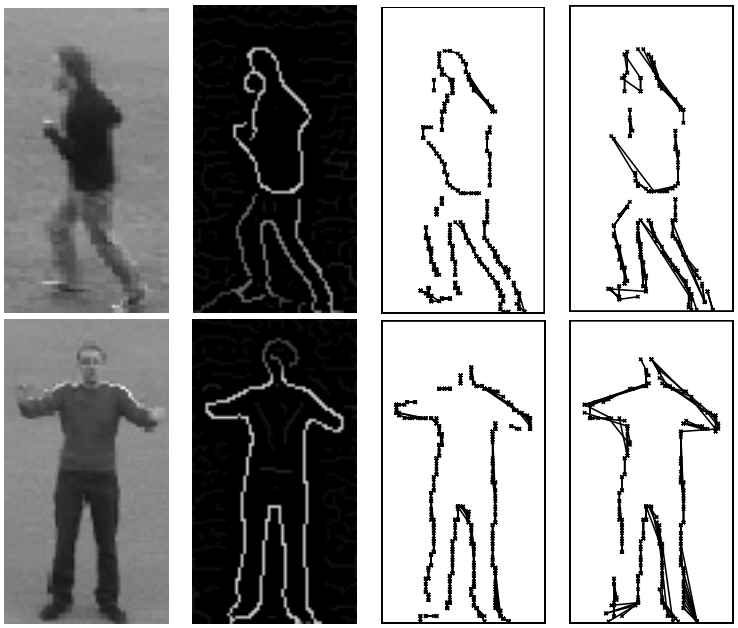
### 3.2.1 Line-based shape features

Shape is an important cue for recognizing the ongoing activity. When we cannot extract the silhouette information from the sequence, due to various reasons like camera movement, zoom effect, etc., we propose to use a compact shape representation based on lines.

We extract this representation as follows: First, given a video sequence, we compute the probability of boundaries (Pb features [58]) based on Canny edges in each frame. We use these Pb features rather than simple edge detection, because Pb features delineate the boundaries of objects more strongly and eliminate the effect of noise caused by shorter edge segments in cluttered backgrounds to a certain degree. Example images and their corresponding boundaries are shown in Fig 3.9(a) and Fig 3.9(b).

After finding the boundaries, we localize the human figure by using the densest area of high response Pb features. We then fit straight lines to these boundaries using Hough transform. We do this in two-fold; first, we extract shorter lines (Fig 3.9(c)) to capture fine details of the human pose. Second, we extract relatively longer lines (Fig 3.9(d)) to capture the coarser shape information.

We then histogram the union of short and long line sets based on their orientations and spatial locations. The lines are histogrammed over $15^\circ$ orientations, resulting in 12 circular bins, similar to the binning of the rectangles in our HOR descriptor. In order to incorporate spatial information of the human body, we evaluate these orientations within a $N \times N$ grid placed over the whole body. Our experiments show that $N = 3$ gives the best results (in accordance with section 3.1). This process is shown in Fig 3.10. Resulting shape feature vector is the concatenation of all bins, having a length $|Q| = 108$ where $Q$ is the set of all features.



(a) (b) (c) (d)

Figure 3.9: Extraction of line-based features. a) The original image. b) Probability of boundary(Pb) features are extracted. c) Short line segments are fitted to the thresholded boundary edges. d) Longer line segments are extracted to capture the more general information about the shape. The final feature vector involves the statistics of both short and long line segments.

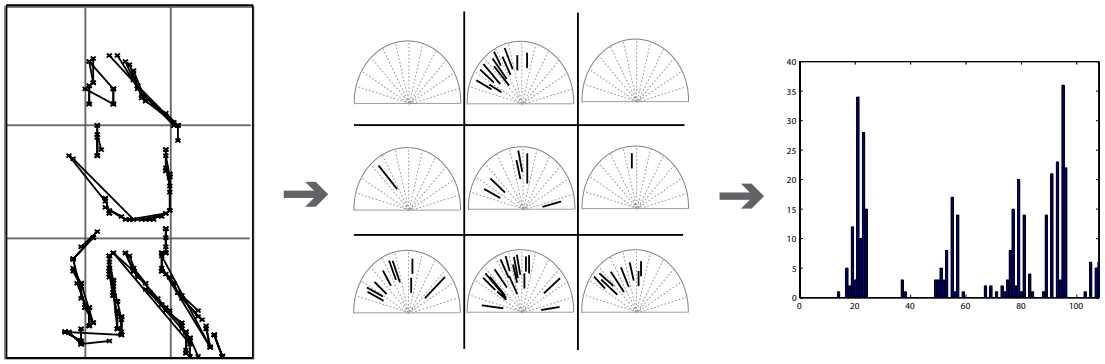


Figure 3.10: Forming line histograms are shown in this figure. An $NxN$ grid is placed over the bounding box of the human figure (here we use $N = 3$) and lines are histogrammed over each spatial grid based on their orientations. The resulting feature vector is the concatenation of all spatial bins.

#### 3.2.1.1 Feature Selection

In our experiments, we observed that, even a feature size of $|Q| = 108$ is a sparse representation for shape. That is, based on the nature of the actions, some of the dimensions of this feature vector are hardly used. To have a more dense and compact representation and to reduce the processing time in classification step, we make use of a maximum-entropy based feature selection approach. By using maximum entropy, we are able to detect regions of interest in which most of the change, i.e motion occurs.

We calculate the entropy of the features as follows: Let $f_j(t)$ represent the feature vector of frame at time $t$ in video $j$ and let $|V_j|$ denote the length of the video. The entropy $H(f_j^n)$ of each feature $n$ over the temporal domain is

$$H(f_j^n) = -\sum_{t=1}^{|V_j|} \hat{f}_j^n(t) log(\hat{f}_j^n(t)) \tag{3.8}$$

where $\hat{f}$ is the normalized feature over time such that

$$\hat{f}_j^n = \frac{f_j^n(t)}{\sum_{t=1}^{|V_j|} f_j^n(t)} \tag{3.9}$$

This entropy $H(f_j^n)$ is a quantative measure of energy in a single feature dimension $n$. A low $H(f_j^n)$ means that the $n$th feature is stable during the action and higher $H(f_j^n)$ means the $n$th feature is changing rapidly in the presence of action. We expect that the high entropy features will be different for different action classes. Based on this observation, we compute the entropies of each feature in all training videos separately for each action. More formally, our reduced feature set $Q'$ is

$$Q' = \left\{ f^n | H(f_j^n) > \tau \ , \ \forall j \in \{1, .., M\}, n \in \{1, .., |Q|\} \right\} \tag{3.10}$$

where $\tau$ is the entropy threshold, $M$ is the total number of videos in training set and $Q$ is the original set of features. After this feature reduction step, our shape feature vector's length reduces to $\approx 30$. Note that for each action, we now have a separate set of features.

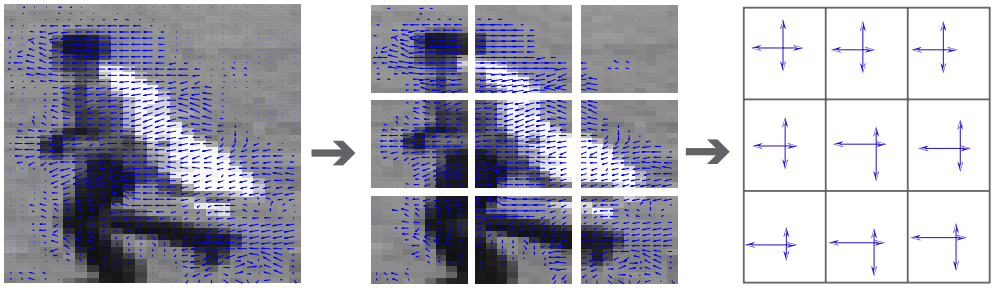


Figure 3.11: This figure illustrates the formation of optical flow histograms. We extract dense block-based OF from each frame. Then, similar to forming shape histograms, we divide the bounding box into equal-sized spatial bins. Inside each bin, we use the total amount of optical flow in four perpendicular directions as our motion descriptor.

### 3.2.2 Motion features

Using pure optical flow (OF) templates increase the size of the feature vector to a great extent. Instead, we present a compact OF representation for efficient action recognition. With this intention, we first extract dense block-based optical flow of each frame, by matching it to the previous frame. We used $L_1$ distance with a block size of $5 \times 5$ and a window size of 3 in this template matching procedure.

We then form orientation histograms of these optical flow values. This is similar to motion descriptors of Efros *et al.* [26], however we use spatial and directional binning instead of using the whole template. In addition, we skip the smoothing step, and use the optical flow values as is. For each $i^{th}$ spatial bin where $i \in \{1, .., N \times N\}$ and direction $\theta \in \{0, 90, 180, 270\}$, we define optical flow histogram $h_i(\theta)$ such that

$$h_i(\theta) = \sum_{j \in B_i} \psi(\tilde{\mathbf{u}}_\theta \cdot \mathbf{F}_j) \tag{3.11}$$

where $F_j$ represents the flow value in each pixel $j$, $B_i$ is the set of pixels in the spatial bin $i$, $\tilde{\mathbf{u}}_\theta$ is the unit vector in $\theta$ direction and $\psi$ function is defined as

$$\psi(x) = \left\{ \begin{array}{ll} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{array} \right\} \tag{3.12}$$

This process is depicted in Fig 3.11.

### 3.2.3 Recognizing Actions

#### 3.2.3.1 SVM classification

After the feature extraction step, we use them for the recognition of actions. We train separate shape and motion classifiers and combine the decisions of these by a majority voting scheme. For this purpose, we again use SVM classifiers. We train separate one-vs-all SVM classifiers for each action. These SVM classifiers are formed using *rbf* kernels over snippets of frames using a windowing approach. In our windowing approach, the sequence is segmented into $k$-length chunks with some overlapping ratio $o$, then these chunks are classified individually (we achieved the best results with $k = 7$ , and $o = 3$).

We combine the vote vectors from the shape $c_s$ and motion $c_m$ classifiers using a linear weighting scheme and obtain the final classification decision in $c_f$, such that

$$\mathbf{c}_f = \alpha\, \mathbf{c}_s + (1 - \alpha)\mathbf{c}_m \tag{3.13}$$

and we choose the action having the maximum vote in $\mathbf{c}_f$. We evaluate the effect of chosing $\alpha$ in the Section 4.3.

#### 3.2.3.2 Including Global Temporal Information

In addition to our local motion information (i.e. OF histograms), we also enchance the performance of our algorithm by using an additional global velocity information. Here, we propose to use a simple feature, which is the overall velocity of the subject in motion. Suppose we want to discriminate two actions: “handwaving” versus “running”. If the velocity of the person in motion is equal to zero, the probability that he is running is quite low.

Based on this observation, we propose a two-level classification system. In the first level, we calculate mean velocities of the training sequences and fit a univariate Gaussian to each action in action set $A = \{a_1..a_n\}$ . Given a test instance, we compute the posterior probability of each action $a_i \in A$ over these Gaussians, and if the posterior

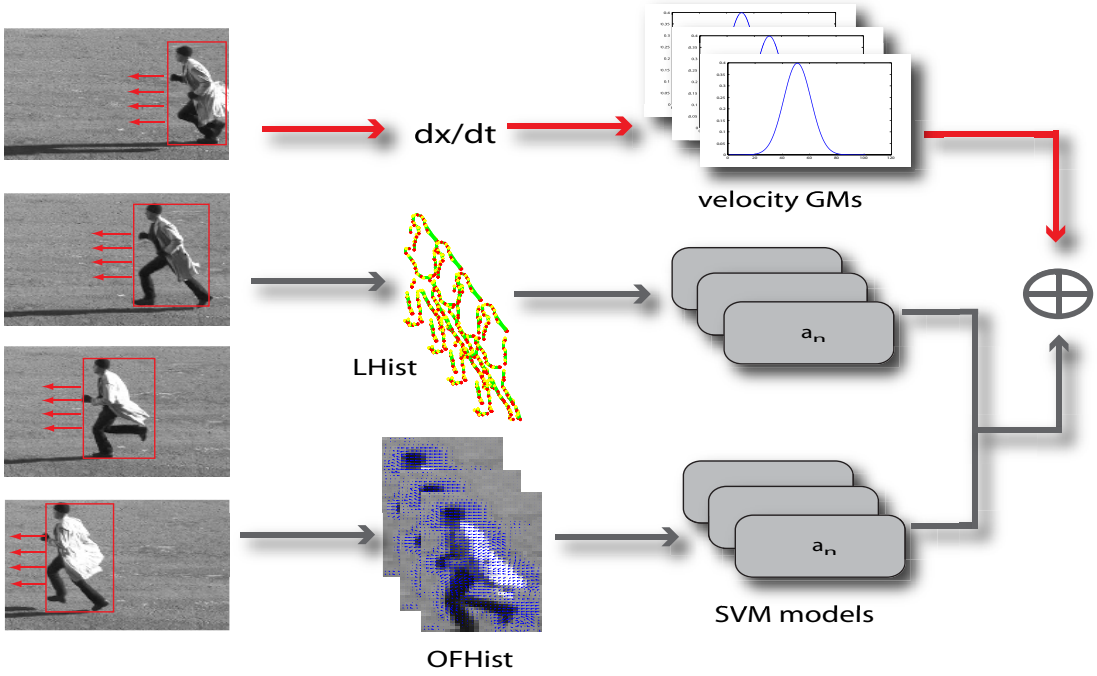



Figure 3.12: Overall system architecture with addition of mean horizontal velocity. Classification votes from line and flow histograms are joint via a linear weighting scheme. The global velocities are modeled using 1D gaussians and the final classification label is achieved by using global velocity as a prior.

probability of $a_i$ is greater than a threshold $t$ (we use a loose bound $t = 0.1$), then we add $a_i$ to the probable set $A'$ of actions for that sequence. After this preprocessing step, as the second level, we evaluate the sequences using our shape and motion descriptor. We take the maximum response of the SVMs for actions $a_k \in A'$ as our classification decision. The overall system is summarized in Fig. 3.12.

## 3.3 Single Action Recognition inside Still Images

Long before the evolution of the video technology, the human actions were conveyed via static images. The newspapers still use action photography to picturize their news. Although motion is a very important cue for recognizing actions, when we look at such images, we can more or less understand human actions in the picture. This is mostly true in news or sports photographs, where the people are in stylized poses that reflect

Figure 3.13: Human mind can perceive the available actions even from a single image, without examining the whole sequence. Here, we show some example images that contaom actions. **Left to right:**Running, walking, kicking, crouching, throwing and catching.

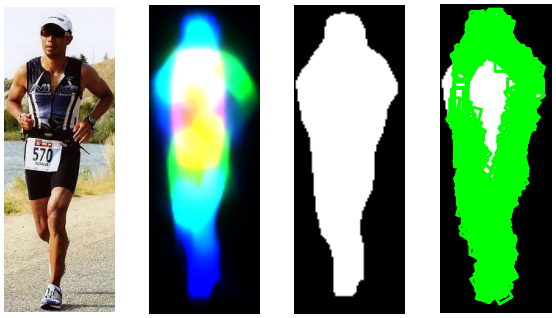
an action. Figure 3.13 shows some example images. However, understanding human actions from still images is a widely ignored problem of computer vision.

In this section, we try to address this problem and answer the question of "Can we recognize human actions within a single image?". Our approach starts with employing a pose extractor, and then representing the pose via distribution of its rectangular regions. By using classification and feature reduction techniques, we test our representation via supervised and unsupervised settings.

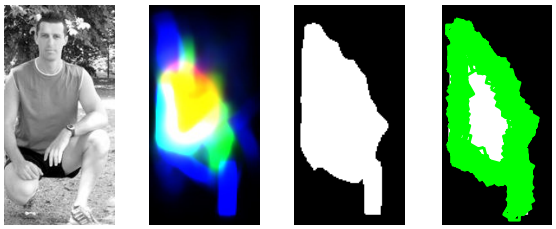
In still images, understanding motion is not a straightforward process. In the presence of motion, it is relatively easier to localize the person, whereas, in still images, we need to estimate the place and pose of the person. However, in the presence of background clutter and occlusions, it is not very straigtforward to localize the person and represent the pose. For this reason, we first use a pose extraction algorithm for estimating the pose of the person in the image. Then, using our shape descriptor, we try to identify the ongoing action. In the remaining of the section, we go into the details of our approach for action recognition in still images.

### 3.3.1 Pose extraction from still images

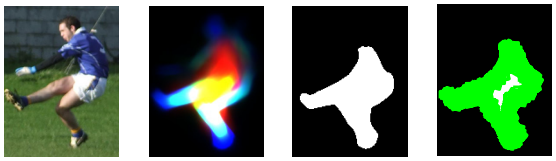
We first use the method of Ramanan [74] to extract a pose from the still image. The approach uses edge and region features, and constructs two deformable models using



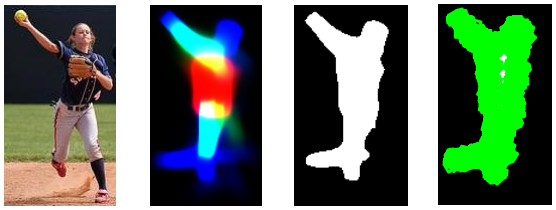
(a) running



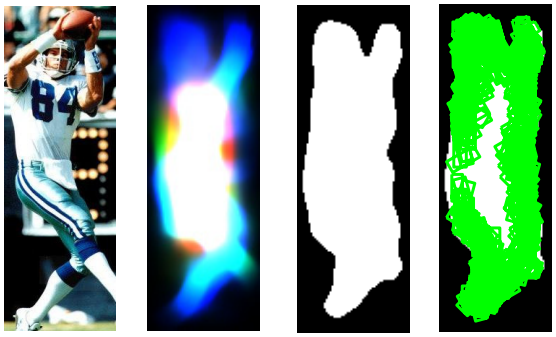
(b) crouching



(c) kicking



(d) throwing



(e) catching

Figure 3.14: Pose and rectangle extraction. **To the left:** The original image and its corresponding parse obtained by using iterative parsing as defined in [74]. **To the right:** The extracted silhouette and the resulting rectangles.

Conditional Random Fields (CRF). Edge-based deformable model consists of K number of parts denoted as $l_i$. Using the part information, the configuration of the model is represented as $L = [l_1, l_2, \ldots l_k]$. This representation is a tree structure, and each part corresponding to a node of the tree has a single parent. The deformable model equation is defined as follows

$$P(L|I)\alpha \exp(\sum_{i,j \in E} \Psi(l_i - l_j) + \sum_i \phi(l_i))$$

Here, $\Psi(l_i - l_j)$, is the priori information of relative arrangements of part $i$ with respect to its parent part $j$. In the study, the shape prior expresses in terms of discrete binning. $\phi(l_i)$ corresponds to local image features extracted from the oriented image patch located at $l_i$. The overall edge-based deformable model is used to estimate the initial body part positions. Then, using the previously obtained estimate, the method creates a region model(parse) that represents an image for each one of the body parts. Then, information obtained from part histograms become the ground for the region-based deformable model. The initial estimates of body positions from region-base model are utilized to build a second region-based model. The procedure continues iteratively by constructing a region model that is based on color evidence.

While pose extraction is still in its infancy, it still gives some idea about the overall posture of the person. Figure 3.14 shows example images and their corresponding poses. In Fig. 3.14(a), an example image and its parse for a “catching” action is shown. As you can see, the background is quite cluttered and the best parse is not very informative in this image. Fig. 3.14(b) gives an example for a “running” action. The parse in this example is more accurate.

We use these initial parses as basis and extract silhouettes by thresholding over the probability maps. Our preliminary experiments have shown that a threshold $\delta = 0.1$ will work the best in this case, because we want to get as many information as possible about the location and shape of the person inside the image. Resulting silhouettes are quite imperfect; for example it is quite difficult for the human eye, to distinguish the silhouette extracted in Fig. 3.14(b) as a “catch” action.

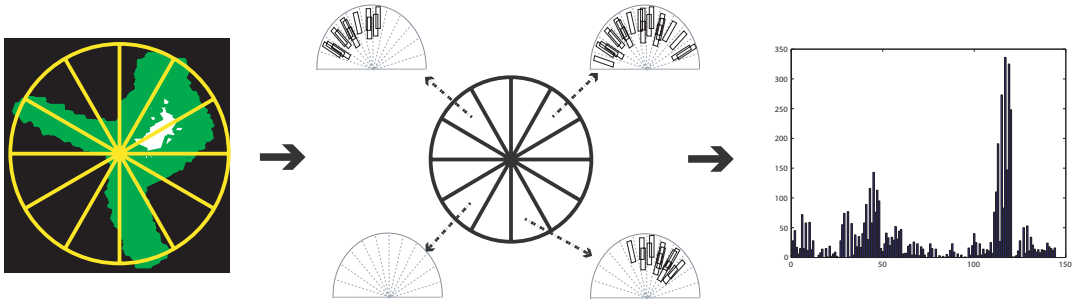


Figure 3.15: Pose representation using circular histogram of oriented rectangles(CHORs). Circular grid is centered to the maximum value of the probability parse.

### 3.3.2 Representing the pose

For describing the human pose, we again make use of rectangular regions that we define in Section 3.1.1. These regions are extracted in the following way: Given the human silhouettes, we search for rectangular regions over this silhouette using convolution of a rectangular filter on different orientations and scales, the same way we do for computing HORs as described in Section 3.1.1. We use undirected rectangular filters, following [76]. The search is performed using 12 tilting angles, which are $15^\circ$ apart. To tolerate the differences in the limb sizes and in the varying camera distances to the subject, we perform the rectangle convolution over multiple scales.

After finding rectangles over the silhouettes, we use Histogram of Oriented Rectangles(HORs [45]) for representing the pose. We compute the histogram of extracted rectangular regions based on their orientations. The rectangles are histogrammed over $15^\circ$ orientations. For still images, we do this histogramming over the spatial circular grids and define circular HORs (CHORs), as opposed to original $N \times N$ grid form. This is mostly because we don't know the explicit location of the human figure. Instead, we use the the center of the highest probability region of the parse as the center of our circular grid. The bins of this circular histogram are $30^\circ$ apart, making 12 bins in total. We depict this process in Fig 3.15.

### 3.3.3 Recognizing Actions in Still Images

As we discuss in Section 3.1.2, HORs are quite compact representations for action recognition in videos. However, we can further densify the representation. In fact, in still image case, we have much lesser examples for action classes, therefore feature reduction is necessary for learning. For this purpose, we first apply Linear Discriminant Analysis(LDA) in our feature space. By using LDA, we reduce the feature dimension from 144 to 50.

We then train one-vs-all SVM classifiers for each action separately and use the highest probable class label. We form the SVM classifiers using RBF kernels.

For evaluating the performance of our method on unsupervised classification, we also apply clustering, and make a qualitative evaluation of clusters. Our clustering procedure is as follows: We run k-means clustering algorithm over the data for $M = 100$ times, and take the clusters that minimize the intra-cluster distance and maximize the inter-cluster distance. The respective results are given in Section 4.4.

# Chapter 4

# Experiments on Single Human Actions

In this chapter, we evaluate the performance of our methods introduced for single action recognition in Section 3. As stated before, by single actions, we refer to the case where the images or videos involve one action only. In the case of single actions, our claim is that, we may not need complex modeling, hence, compact shape representations may suffice for identifying the human actions. In this chapter, we exploit the dimensions of our claim and we further boost the classification performance by combining our shape features with temporal motion features. We show that we can achieve high-accuracy action recognition with the help of our compact spatio-temporal features.

## 4.1 Datasets

We test the effectiveness of our method over two state-of-the-art datasets over the video domain. The first dataset is the Weizzman dataset and second is the KTH dataset; these are the current benchmark datasets in the action recognition research. For computing the performance of our approach for action recognition in still images, we make use of two distinct datasets, one is the ActionWeb dataset, second is the figure skating dataset [100]. We now describe these datasets in greater detail.

Figure 4.1: Example frames from the Weizzman dataset introduced in [12].

### 4.1.1 Video Datasets

**Weizzman dataset:** This is the dataset that Blank *et al.* introduced in [12]. We use the same set of actions as in [12], which is a set of 9 actions: walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place and jumping jack. Example frames from this dataset are shown in Fig. 4.1. We used the extracted masks provided to localize the human figures in each image. These masks were obtained using background subtraction. We test the effectiveness of our method using leave-one-out cross validation.

**KTH dataset:** This dataset has been introduced by Schuldt *et al.* in [86]. It is more challenging than the Weizzman, covering 25 subjects and 4 different recording conditions of the videos with the frame rate of 25fps and a spatial resolution of $160 \times 120$. There are 6 actions in this dataset: boxing, handclapping, handwaving, jogging, running and walking. One additional challenge of this dataset comes from the set of actions available; there are two very similar actions – jogging and running – in this dataset. Example frames from the KTH dataset are shown in Fig. 4.2. There are four different recording conditions of this dataset, where each actor and each action is recorded repeatedly. These are: **s1:** the standard outdoor recording with a stable camera(Fig. 4.2(a)), **s2:** zoom-effect of the camera for boxing, handwaving and handclapping actions, and diagonal movements (i.e. different viewpoints) for running, jogging and walking actions(Fig. 4.2(b)), **s3:** actors with different outfits and carry items with

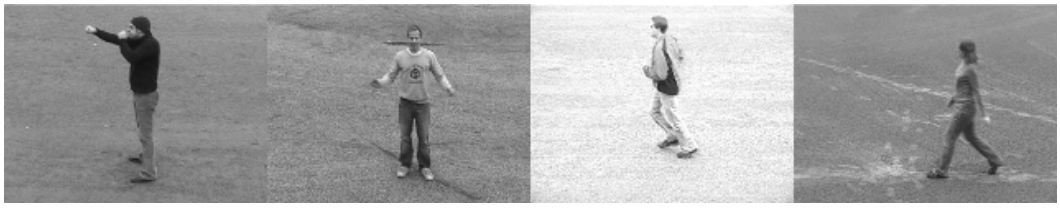
more strong shadow effects(Fig. 4.2(c)), **s4:** indoor(darker) recording with considerable amount of illumination change and shadows(Fig. 4.2(d)). Video sequences are in the order of 100 frames, and the actions are repeated multiple times in each video. Since there is no change in the nature of the action throughout the whole video, processing only first 40 frames–where the actor is in view of the camera– is sufficient for our experiments.

Since the recording conditions of the videos in the KTH dataset are not stable, and there is considerable amount of camera movement in some situations, silhouette extraction in this dataset is not straightforward. For this reason, we make use of several clues like lines and gradients, for a good extraction of the foreground human figure. In the KTH dataset, despite the camera movement and zoom effect, the backgrounds of the sequences are relatively simple. We used this fact to localize the human figure, and then applied background subtraction to the localized image region to extract the silhouettes. We perform localization based on the density of the vertical lines and gradients. The resulting silhouettes are not perfect, but realistic. Example silhouettes are given in Figure 4.3. The successful results on these noisy silhouettes prove that our method does not heavily depend on perfect silhouettes. We should note that, better silhouettes will give higher accuracy rates eventually.

### 4.1.2 Still Image Datasets

**ActionWeb dataset:** For recognition of actions from still images, we collected a dataset from various sources like Google Image Search, Flickr, BBC Motion database, etc. This dataset consists of 467 images and includes six different actions; these are running, walking, catching, throwing, crouching and kicking. We choose this subset of actions, because these are mostly visually identifiable actions from single images. Example images for each action is shown in Fig 4.4. This image collection involve a huge amount of diversity by means of viewpoints, shooting conditions, cluttered backgrounds, resolution.

**Figure Skating dataset:** We also test our descriptor's performance for the case of unsupervised classification. For this purpose, we used Wang *et al.*'s skating images



(a) s1 condition: outdoor(standard recording)

(b) s2 condition: zoom effect and different viewpoints

(c) s3 condition: different outfits and carry items

(d) s4 condition: indoor(darker recording)

Figure 4.2: Example frames from the KTH dataset introduced in [86].

(a) boxing silhouettes

(b) handclapping silhouettes

(c) handclapping silhouettes

(d) jogging silhouettes

(e) running silhouettes

(f) walking silhouettes

Figure 4.3: Extracted silhouettes from the KTH dataset in s1 recording condition. Each single silhouette image corresponds to a different actor in the dataset (a total of 25 actors). The silhouettes are quite imperfect, due to the difficulty in foreground/background discrimination in this dataset.

(a) running

(b) walking

(c) throwing

(d) catching

(e) crouching

(f) kicking

Figure 4.4: Example images of the ActionWeb dataset collected from the web sources. There is a large amount of diversity in this dataset, with respect to pose, pose direction, background clutter and scale.

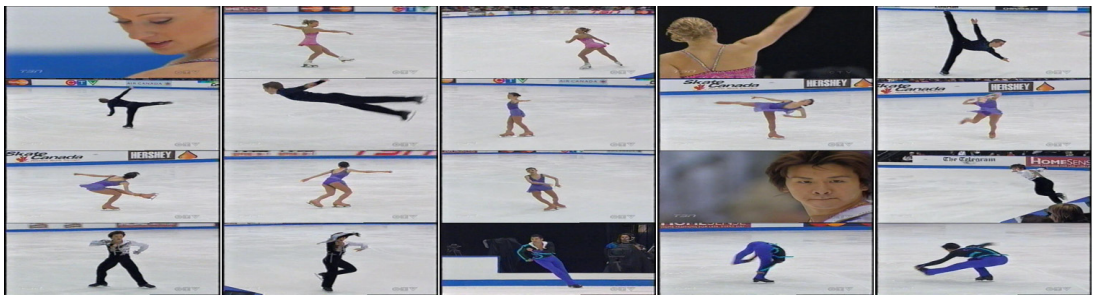


Figure 4.5: Example frames from the figure skating dataset introduced in [100].

dataset [100]. This dataset is a collection of 1432 images, where different figure skaters perform various moves. Example images from this dataset is given in Fig. 4.5. The existing categories of this figure-skating dataset can be generalized under 10 different labels: face close-up picture, skates with arms down, skates with one arm out, skater leans to his right, skates with both arms out, skates on one leg, sit-spin leg to the left of the image, sit spin leg to right of the image, camel spin leg to left of the image and camel spin leg to right of the image.

## 4.2 Experiments with Histogram of Oriented Rectangles (HORs)

### 4.2.1 Optimal Configuration of the Pose Descriptor

In this section, we present the experiments regarding the performance of the histogram-of-oriented-rectangles(HOR) pose descriptor with respect to its configuration. There are several choices that can be made while forming the HOR pose descriptor. These are (a) granularity of the angular bins, i.e. number of orientations for the rectangle detection, (b) number of spatial bins and (c) choice of torso exclusion.

In the following, we first exploit the effect of using different configurations over the Weizzman data set. Based on this evaluation, we determine the optimal configuration and continue the rest of the experiments using this optimal configuration.

Table 4.1: The accuracies of the matching methods with respect to angular bins (over a grid of $3 \times 3$. The original rectangle search is done with $15^\circ$ tilted rectangular filters. To form $30^\circ$ histograms, we group rectangles that fall into the same angular bins. These results demonstrate that as we move from fine to coarser scale of angles, there is a slight loss of information, and thus $30^\circ$ HORs become less discriminative than $15^\circ$ HORs. $180^\circ$ HORs ignore the orientation information of the rectangles and performs binning based on the spatial distribution of the rectangles over the silhouette. Surprisingly, even the spatial distribution of the rectangular regions provide quite rich information about the available action.

| Classification Method | $15^\circ$ | $30^\circ$ | $180^\circ$ |
|---|---|---|---|
| NearestNeighbor | 96.30% | 95.06% | 92.59% |
| GlobalHist | 96.30% | 93.83% | 85.19% |
| SVM | 97.53% | 93.83% | 92.59% |
| DTW | 100% | 95.06% | 91.36% |

#### 4.2.1.1 Granularity of Angular Bins

We first evaluate the choice of orientation angles when forming the histogram. Table 4.1 shows the results using different angular bins. The original rectangle search is done with $15^\circ$ tilted rectangular filters. To form $30^\circ$ histograms, we group rectangles that fall into the same angular bins. Not surprisingly, we see that there is a slight loss of information when we go from fine level orientations (i.e. $15^\circ$ bins) to a coarser level ($30^\circ$). More interestingly, if we do not use angular binning and instead use just the histogram of rectangles falling into each spatial grid, we still capture a valuable amount of information ($180^\circ$ case). $180^\circ$ HORs ignore the orientation information of the rectangles and performs binning based on the spatial distribution of the rectangles over the silhouette. This confirms that describing the human figure as a collection of rectangles is a sensible approach, and even the spatial distribution of the rectangular regions over the silhouette provide quite rich information about the available action. If we look at the orientation of these rectangles besides the spatial distributions, we acquire more detail and higher accuracy about the action of the body.

Table 4.2: The accuracies of the matching methods with respect to $N \times N$ grids (with $15^\circ$ angular bins, no rectangle or torso elimination). We have compared $2 \times 2$ and $3 \times 3$ partition grids. Our results show that the $3 \times 3$ grid is more effective when forming our oriented-rectangles based pose descriptor.

| Classification Method | $1 \times 1$ | $2 \times 2$ | $3 \times 3$ |
|---|---|---|---|
| NearestNeighbor | 64.20% | 91.36% | 96.30% |
| GlobalHist | 55.56% | 87.65% | 96.30% |
| SVM | 80.25% | 90.12% | 97.53% |
| DTW | 70.37% | 91.36% | 100% |

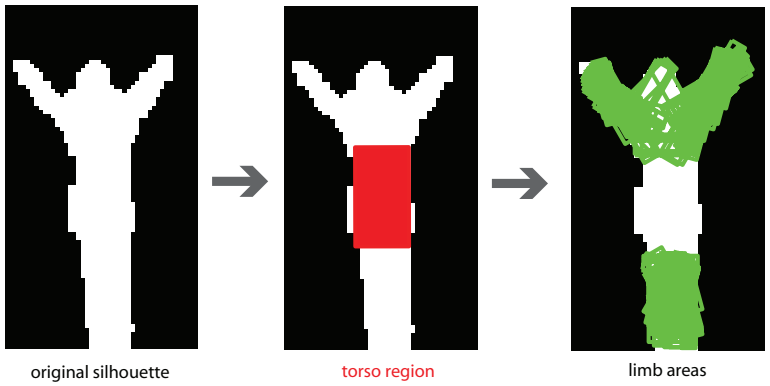
#### 4.2.1.2 Number of Spatial Bins

When forming the histograms of oriented rectangles, we place an $N \times N$ grid over the silhouette of the subject and form orientation histograms for each grid region. The choice of $N$ effects the size of the feature vector (thus execution time of the matching), and the level of detail of the descriptor. Table 4.2 compares the use of different spatial grids. The $1 \times 1$ grid implies that we do not use any spatial binning and we take the silhouette as a whole. Not surprisingly, ignoring the spatial layout and binning only over orientations is not satisfactory, since spatial layout of the rectangles provides useful cues for discriminating different parts of the body.

Our results over this dataset indicate that the $3 \times 3$ grid gives a better performance compared to $2 \times 2$. However, if execution time is crucial, choice of $N = 2$ will still work to a certain degree of performance. One can try further levels of partitioning, even form pyramids of these partitions. However, overly dense partitioning will not make sense, since the subregions have to be large enough to contain a reasonable amount of rectangular patches.

#### 4.2.1.3 Torso Detection

One can also perform a special search for the torso rectangle, which is considerably larger than limb rectangles, omit this torso region while searching for the remaining




(a)

| Classification Method | No Torso | Torso |
|---|---|---|
| NearestNeighbor | 96.30% | 96.30% |
| GlobalHist | 96.30% | 91.36% |
| SVM | 97.53% | 95.06% |
| DTW | 100% | 98.77% |

(b)

Figure 4.6: Rectangle detection with torso exclusion (best viewed in color). In (a), the torso region is detected. This is done by applying a larger rectangular filter and taking the mean of the responses. After finding the torso, the remaining silhouette is analyzed for candidate limb areas. In (b), the accuracies of the matching methods with respect to torso exclusion are given (using $15^\circ$ angular bins and $3 \times 3$ grid). We can say that torso detection degrades the performance, so using the whole silhouette for candidate rectangle regions results in higher performance.

body parts and then form rectangular histograms. An example case for this kind of rectangular search is given in Fig. 4.6. Here, by applying a larger rectangular filter and computing the mean among the responses, we localize the torso region. Then, we exclude this region and apply rectangle filtering in order to find candidate limb areas and base our pose descriptor on those remaining areas only.

In Fig. 4.6(b), we show the effect of torso detection on the overall accuracies. We observe that with global histogramming methods, torso detection and exclusion helps; however, SVM and DTW classifiers suffer from slight performance degradation. So, we conclude that explicit torso detection is not necessary and extracting the HOR descriptors from the whole silhouettes is more informative.

Table 4.3: Overall performance of the matching methods over the Weizzman and KTH datasets. Here, *v+SVM* refers to using SVM classifiers together with global velocity Gaussians, and *v+DTW* corresponds to using DTW with the same set of Gaussians.

| Classification Method | Feature | Weizzman | KTH |
|---|---|---|---|
| NearestNeighbor | HOR | 96.30% | 75.46% |
| | HORW | 97.53% | 72.22% |
| GlobalHist | HOR | 96.30% | 71.76% |
| | HORW | 69.14% | 57.41% |
| SVM | HOR | 97.53% | 77.31% |
| | HORW | 95.06% | 85.65% |
| DTW | HOR | **100%** | 74.54% |
| | HORW | 96.30% | 78.24% |
| v+SVM | HOR | 98.77% | 81.48% |
| | HORW | 95.06% | **89.35%** |
| v+DTW | HOR | **100%** | 81.02% |
| | HORW | 98.77% | 83.8% |

### 4.2.2 Classification Results and Discussions

After deciding on the optimal configuration of the pose descriptor, we evaluate the effect of using different classification techniques. We use the optimal configuration found in the previous section which is a $3 \times 3$ grid over $15^\circ$ angular bins as our HOR configuration.

The overall results over two datasets are shown in Table 4.3. For the Weizzman dataset, where actions are mostly differentiable based on their shape information, applying DTW over HOR descriptors gives the best results. However, on the more complex KTH dataset, we need to make use of the velocity information, because shape is mostly not enough, especially in the presence of noise introduced by imperfect silhouettes. In the KTH dataset, best results are achieved by using two level classification with SVM models, which is the global velocity plus SVM classification(v+SVM) using HORWs as features.

In Fig. 4.7 and Fig. 4.8, we present the confusion matrices of our method in Weizzman and KTH datasets respectively. On the Weizzman dataset we achieve the

best results with DTW matching. This is not surprising, because the subjects do not perform actions with uniform speeds and lengths. Thus, the sequences need aligning. DTW matching accomplishes this alignment over the bins of the histogram separately, making alignment of limb movements also possible. Action speed differences between body parts are handled this way.

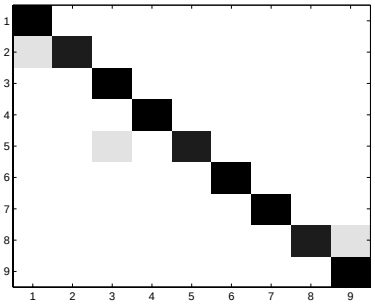
However, in the KTH dataset, a simple alignment using DTW is not sufficient, because in DTW we lose valuable temporal information by warping the time axes while trying to aligh the sequences, and only pay attention to the ordering of the poses. The KTH dataset introduces additional challenge by including very similar actions like jogging and running, which need temporal features to achieve better separation. Therefore, in this dataset, v+SVM classification performs best.

We should also note that, especially on the Weizzman dataset, nearest neighbor voting per frame and global histogramming with our pose descriptor produce surprisingly good results. This suggests that we can still achieve satisfactory classification rates even if we ignore the time domain to a certain degree and look at the frames separately, or as a whole.

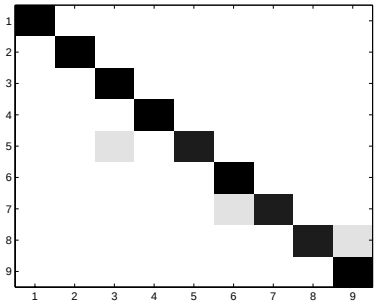
One shortcoming of our rectangle-based approach is its dependence over silhouette extraction. We observed that most of the confusion, especially in the KTH dataset, occurs because of the imperfect silhouettes. However, we should also note that, even with imperfect silhouettes, our method achieves high recognition rates which shows our method's robustness to noise. We argue that better silhouettes will result in higher accuracy rates eventually.

### 4.2.3 Comparison to other methods and HOGs

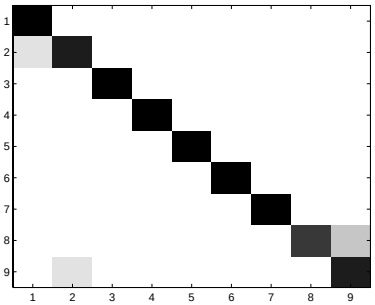
We reach a perfect accuracy (100%) over the Weizzman action dataset, using $15^\circ$ angular bins over a $3 \times 3$ spatial partitioning with DTW or v+DTW methods. We present comparison of our results over this dataset in Table 4.4. Blank *et al.* report classification error rates of 0.36% and 3.10% for this dataset. Recently, Niebles and Fei Fei [65] evaluate their hierarchial model of spatial and spatio-temporal features over



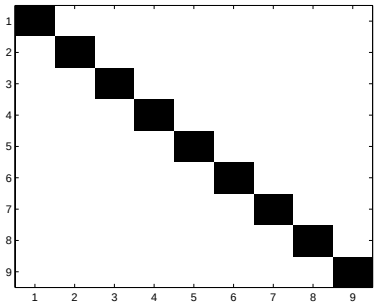

(a) Nearest neighbor: 1 jump sequence classified as bend, 1 one-hand wave sequence classified as jump-in-place and 1 run sequence misclassified as walk.




(b) Global histogramming: 1 one-hand wave sequence misclassified as jump-in-place, 1 jumpjack sequence misclassified as two-hands-wave and 1 run sequence misclassified as walk.




(c) SVMs: 1 jump sequence is classified as bend, 2 run sequences classified as walk, 1 run sequence misclassified as jump.




(d) DTW method achieves 100% accuracy.

Figure 4.7: Confusion matrices for each matching method over the Weizzman dataset with the optimal configuration.

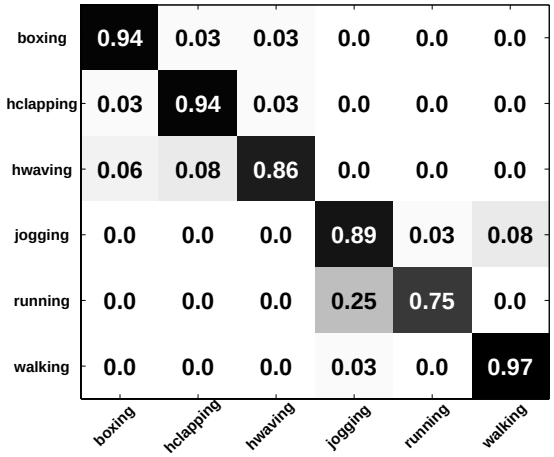



Figure 4.8: Confusion matrix for classification results of the KTH dataset using HORW feature with v+SVM. The overall accuracy we achieve is 89.35% in this dataset. Most of the confusion occurs between run and jog actions, which is quite comprehensible.

Table 4.4: Comparison of our method to other methods that have reported results over the Weizzman dataset.

| Method | Accuracy |
|---|---|
| **HOR** | **100%** |
| Blank et al. [12] | 99.64% |
| Jhuang et al. [49] | 98.8% |
| Wang et al. [99] | 97.78% |
| Niebles et al. [65] | 72.8% |

this dataset, acquiring an accuracy of 72.8% and Wang and Suter [99] used FCRF over a grid-based feature, resulting in an accuracy of 97.78%.

In Table 4.5, we compare our descriptor's performance to current results on the KTH dataset. We should note that the numbers in Table 4.5 are not directly comparable, because the testing settings are different. Some of the approaches use leave-one-out cross-validation, whereas some others use different splitting of train and test data. We use the train and test sets provided in the original release of the dataset by Schuldt *et al.* [86]. Overall, we can say that our approach is among the top-ranking approaches in the literature regarding this dataset.

We also compare our approach to the Histogram of Oriented Gradients(HOG),

Table 4.5: Comparison of our method to other methods that have reported results over KTH dataset.

| **Method** | **Accuracy** |
|---|---|
| Jhuang et al. [49] | 91.7% |
| Wong et al. [103] | 91.6% |
| **HORW** | **89.4%** |
| Niebles et al. [66] | 81.5% |
| Dollár et al. [25] | 81.2% |
| Ke et al. [51] | 80.9% |
| Schuldt et al. [86] | 71.7% |

which is also based on histogramming and therefore is a natural counterpart to our approach. The HOG method has been recently proposed by Dalal and Triggs [22]. They have used gradient orientations to detect humans in still images, and their approach has been shown to be quite successful.

We used provided HOG implementation in order to extract the HOGs in the KTH dataset. While doing this, we omit the human detection phrase and we compute HOG features directly over the bounding box of the extracted silhouettes, using parameters $cellsize = 8$ and $\#ofcells = 2$. This gives a feature vector of size 288, which is computationally very expensive, especially when used with SVMs over window of frames. In order to cope with this, and to be more computationally efficient, we reduce the size of the HOG vectors by applying PCA and using the projections over the 80 principal components.

Table 4.6 shows the comparison results of HOGs and HORs using three of the most successful matching methods over the KTH dataset. As the results indicate, using HORs as opposed to HOGs gives a better classification performance in all matching techniques.

Table 4.6: Comparison to HOG feature based action classification over the KTH dataset.

| | HOG | HOR | HORW |
|---|---|---|---|
| SVM | 76.85% | 77.31% | 85.65% |
| DTW | 67.59% | 74.54% | 78.24% |
| v+SVM | 82.41% | 81.48% | 89.35% |

Table 4.7: Run time evaluations for different matching techniques using HORs. The results presented here are the per-frame running times over the Weizzman dataset.

| | NearestNeighbor | GlobalHist | SVM | DTW | v+SVM | v+DTW |
|---|---|---|---|---|---|---|
| msec | 70.58 | 3.82 | 32.0 | 81.84 | 35.49 | 82.47 |

### 4.2.4 Computational Evaluation

The run-time evaluation of our approach is two-fold. First is the phase of rectangle extraction. Rectangle extraction consumes around 1sec per frame.

The second phase is the matching part. The computational evaluation of the methods (implemented in MATLAB without code optimization) is presented in Table 4.7. These results are the running times (per frame) of corresponding methods over the Weizzman dataset. DTW is the most time-consuming method among others, whereas global histogramming takes the least amount of time. SVM classification has very managable time constraints and is preferable if the running time is an important consideration of the system.

## 4.3 Experiments with Line and Flow Histograms

When the silhouette information is not extractable, but the image sequences have relatively simple backgrounds and the foreground human figure is identifiable, we can use boundary-fitted lines instead of rectangles. In this section, we present experimental

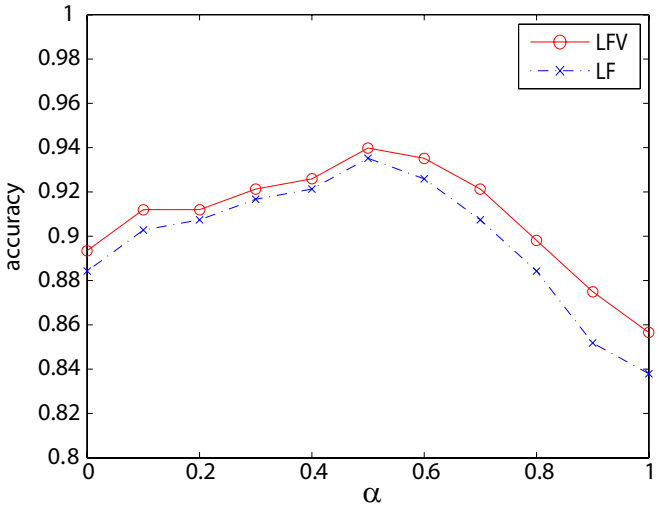



Figure 4.9: Choice of $\alpha$ and resulting confusion matrix for the KTH dataset. $\alpha = 0$ means that only motion features are used, whereas $\alpha = 1$ corresponds to using line-based shape features alone. We see that the mutual combination of these two features, with equal weights, gives the best classification accuracy.

evidence for supporting this claim.

In Fig 4.9, we show the performance of our line and flow histograms. Here, LF corresponds to using line and flow histograms without the velocity information, and LFV is with global velocity. We first show the effect of adding global velocity information. We observe that using global information gives a slight improvement on the overall accuracy. In the same figure, we also evaluate the effect of choosing $\alpha$ of Eq. 3.13 which is the weight used for combining line and flow features. In this figure, $\alpha = 0$ indicates that only motion features are used, whereas $\alpha = 1$ corresponds to using only shape features. Our results show that $\alpha = 0.5$ gives the best combination. This coincides with the observations of Ke *et al.* [51], that the shape and motion features are complimentary to each other.

The respective confusion matrix is shown in Fig 4.10. Not surprisingly, most of the confusion occurs between jog and run actions which are very similar in nature.

Here, we should also note that our optical flow feature is quite successful over this dataset. Even used solely, i.e. when $\alpha = 0$, the prediction accuracy we get is 89.35% with SVM classification.

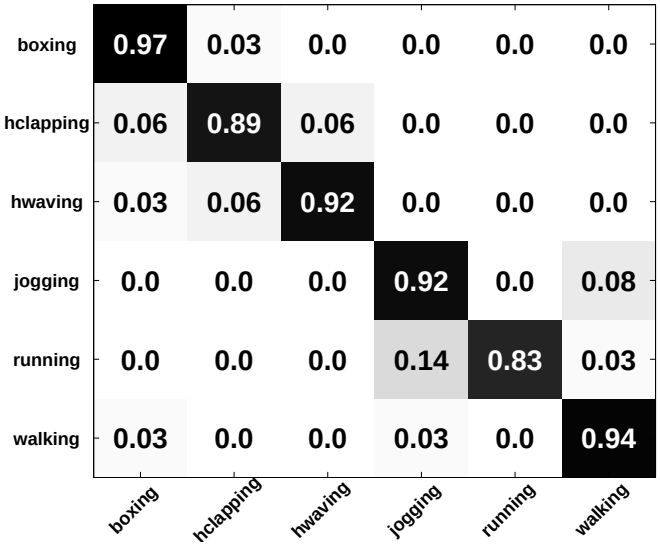



Figure 4.10: Resulting confusion matrix for the KTH dataset. The overall classification accuracy we achieve is 94.0% when we use line and optical flow histograms together with global velocity posteriors.

Table 4.8: Comparison of our method to other methods on KTH dataset.

| Method | Accuracy |
|---|---|
| **LFV** | **94.0%** |
| Jhuang [49] | 91.7% |
| Wong [103] | 91.6% |
| Niebles [66] | 81.5% |
| Dollár [25] | 81.2% |
| Ke [51] | 80.9% |
| Schuldt [86] | 71.7% |

Table 4.9: Comparison with respect to recording condition of the videos in the KTH dataset.

| Condition | LFV | Jhuang [49] |
|---|---|---|
| s1 | **98.2%** | 96.0% |
| s2 | **90.7%** | 86.1% |
| s3 | 88.9% | **89.8%** |
| s4 | **98.2%** | 94.8% |

In Table 4.8, we compare our method's performance to all major results on the KTH dataset reported so far (to the best of our knowledge). We achieve the highest accuracy rate (94%) on this state-of-art dataset, which shows that our approach successfully discriminates action instances. This is the best accuracy obtained for KTH dataset in the literature. We also present accuracy rates for different recording conditions of the dataset in Table 4.9. Our approach outperforms the results of [49] in three out of four of the conditions. Although still very close to the current best results in the literature, the lowest prediction accuracy is in s3 condition, which is not surprising. The presence of carry items and different outfits in this recording condition alter the histogram statistics inevitably, thus, degrading the performance. In such cases, an individual modelling of carry items and non-shape-conserving outfits would help.

Without feature selection, the total classification time (model construction and testing) of our approach is 26.47min. Using feature selection, this time drops to 15.96min. As expected, we gain considerable amount of time as we use a more compact feature representation.

## 4.4 Experiments on Still Images

While searching for the answer to the question "Can we recognize actions from still images?", we applied our CHOR pose descriptor to the extracted parses from the still

images. For performance measurement, we use ActionWeb dataset introduced in Section 4.1.2. We follow leave-one-out cross validation scheme in our experimental evaluation over this dataset.

Figure 4.11 shows examples for the correctly classified images by our approach. Note that the diversity of the images in the dataset is very large, with cluttered backgrounds, different poses, outfits and also carry items.

We also provide examples for the misclassified images in Fig. 4.13. The misclassification labels are written below the images. It can be observed that within the misclassified images, the lack of proper edge boundaries make the pose extraction harder. Additionally, some of the poses are very similar, indistinguishable even to the human eye.

Our overall accuracy rate for supervised classification on ActionWeb dataset is 85.1%. This is a surprisingly good result, given the fact that the images cover a wide range of poses (see Fig. 3.13) and foreground parses are not that perfect (Fig. 3.14). However, by using CHORs, these results show that we can still overcome most of such discrepancies and achieve high accuracy rates.

The resulting confusion matrix for our method with supervised classification over the ActionWeb dataset is given in Fig. 4.12. Most of the confusion occurs between catching and running motions, and this is mostly due to the nature of the photographs. Most of the catching photographs contain running as a sub-action, that is the photographs mostly illustrate the joint of a “run-catch” action. Thus, when the parses miss out arm information, the remaining is a good match for a run action. We can see that our method will benefit a lot from the improvements in body parse estimation techniques.

We also present qualitative results of clustering with our approach. Figure 4.14 presents some of the clusters that we get with the Wang *et al.*’s dataset. We used $k = 100$ in our execution of k-means clustering. As seen, the clusters we get are quite coherent and each of them represents a certain pose.

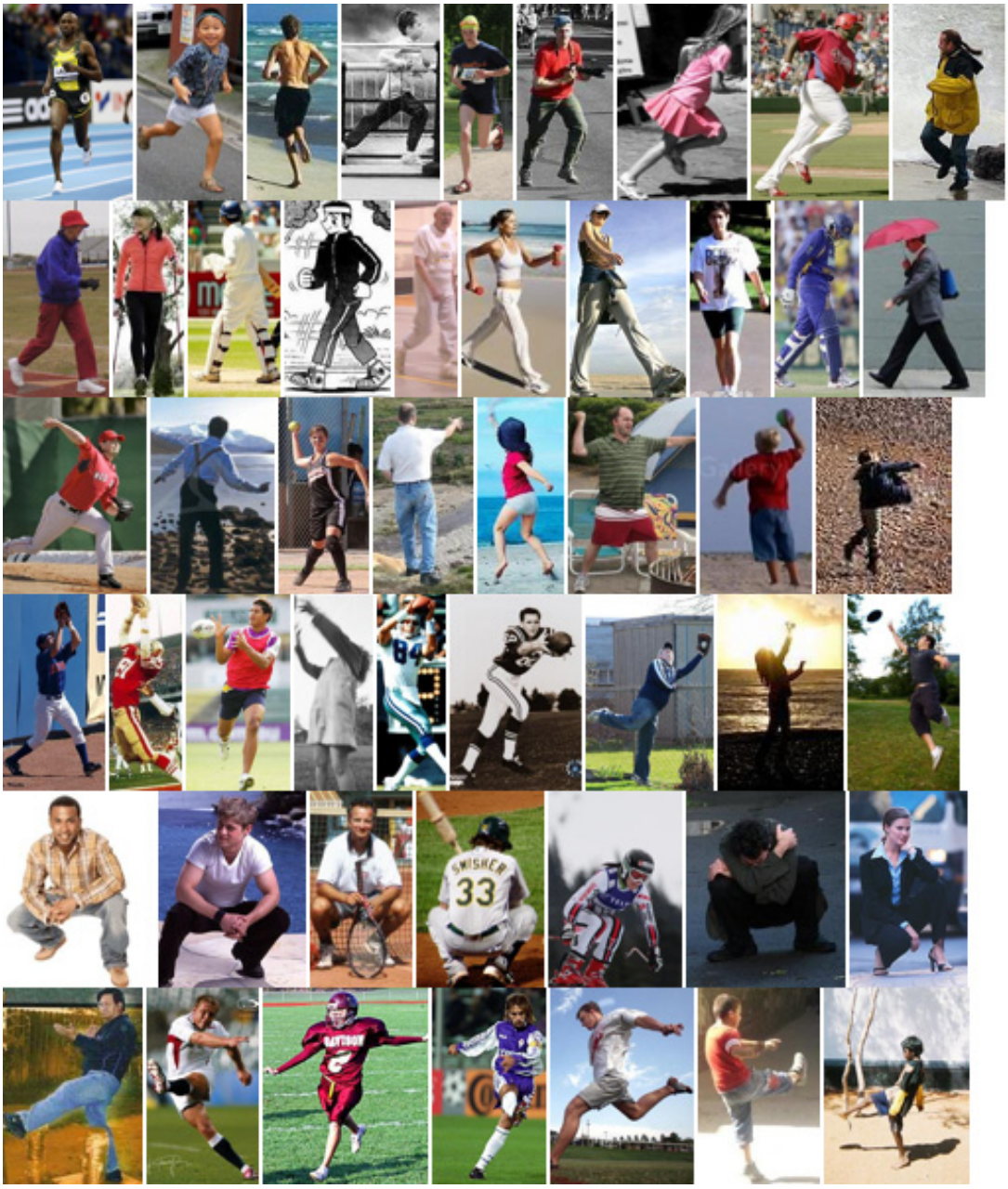


Figure 4.11: Examples for correctly classified images of actions running, walking, throwing, catching, crouching, kicking in consecutive lines.

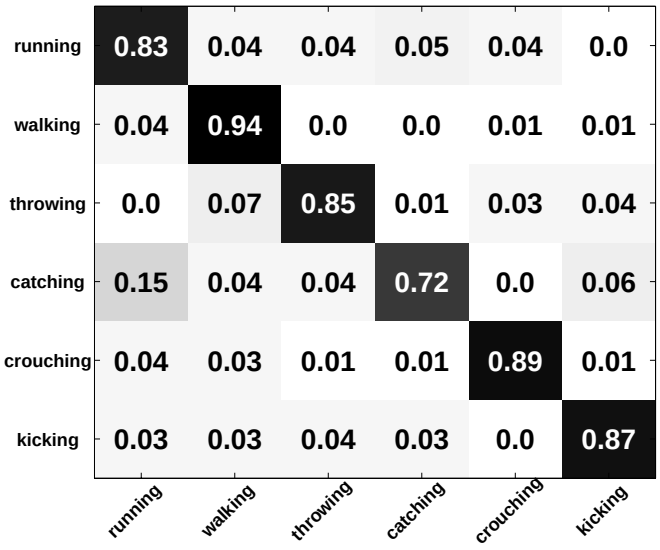
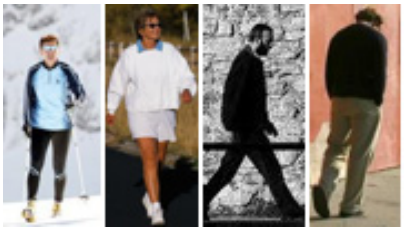



Figure 4.12: Confusion matrix of CHOR method over the ActionWeb still images dataset. The overall accuracy with CHOR pose descriptor for ActionWeb dataset is 85.1%. Most of the confusion occurs between catching and running motions. This is comprehensible, when we examine the nature of the photographs. Most of the catching photographs contain running as a sub-action, illustrating a joint of “run-catch” action.

(a) catch, walk, catch, throw



(b) run, run, run, kick

(c) catch, kick, walk, crouch

(d) run, throw, run, run

(e) kick, walk, walk, catch

(f) throw, walk, run, throw

Figure 4.13: Examples for misclassified images of actions running, walking, throwing, catching, crouching, kicking, consecutively with their wrong classification labels.

Figure 4.14: Clusters formed by our approach for the figure skating dataset. Each row represents a different cluster formed by applying kmeans over CHORs.

# Chapter 5

# Recognizing Complex Human Activities

In general, videos involve more than single actions. The people in motion may exhibit composite activities, where the arms are involved in one action, and the legs are involved in another action. In addition, the actions may be composed over time, forming sequential activities. Furthermore, the direction of the actions may change rapidly, resulting in different viewpoints and shooting angles of the ongoing activity. 2D models which are trained over single viewpoints are most likely to fail under these circumstances.

In such cases, where simple approaches do not suffice, we need more sophisticated models in order to grasp activity information. This chapter introduces such an approach, where the body is modelled in 3D, by means of authoring compositional activity models to distinct limbs separately. Here, we propose to make use of motion capture data to overcome the data shortage problem. It is interesting to note that this data does not consist of everyday actions, but rather a limited set of American football movements. Using motion capture dataset gives us the flexibility of extending our training set and learning from a different domain, and transfering this knowledge to everyday-action recognition domain. This is a type of transfer learning problem, and we believe that transfer learning helps a lot for understanding human activities.

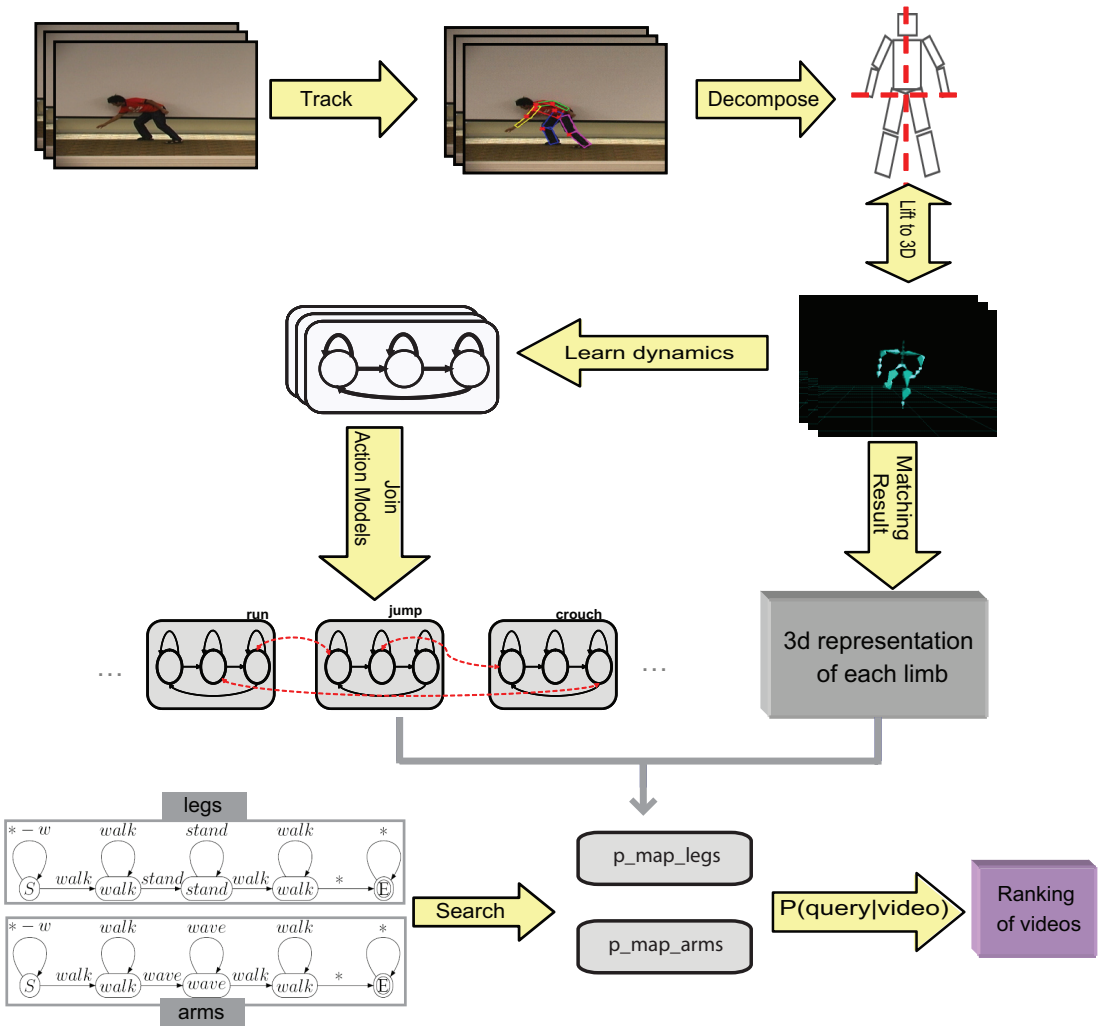



Figure 5.1: Overall system architecture for the retrieval of complex human activities.

Tracking is now a usable, if not perfect technology (section 5.2). Building extremely complex dynamical models from heterogenous data is now well understood by the speech community, and we borrow some speech tricks to build models from motion capture data (section 5.1) to minimize parameter estimation.

Figure 5.1 summarizes the overall system architecture.

## 5.1 Representing Acts, Actions and Activities

In terms of acts and activities, there are many quite different cases. Motions could be sustained (as in walking, running motions) or have a localizable character (as in catching, kicking). The information available to represent what a person is doing depends on timescale. In our study, we distinguish between short-timescale representations (**acts**), like a forward-step; medium timescale **actions**, like walking, running, jumping, standing, waving, whose temporal extent can be short (but may be long) and are typically composites of multiple acts; and long timescale **activities**, which are complex composites of actions.

In order to handle activities, we start with building separate models for actions. Since we want our complex, composite activities to share a vocabulary of base units, we use the kinematic configuration of the body as distinctive feature.

We want our representation to be as robust as possible to view effects and to details of appearance of the body. Furthermore, we wish to search for activities without possessing an example. All this suggests working with an inferred representation of the body's configuration (rather than, say, image flow templates as in [26, 12]). An advantage of this approach is that models of activity, etc. can be built using motion capture data, then transferred to use on image observations, and this is what we do.

Here, we ignore limb velocities and accelerations because actions like reach/wave can be performed at varying speeds. This is mostly true in our case where we use motion capture dataset of American football movements. In this dataset, the players tend to perform actions in higher velocities and accelerations. Our model would have a higher value of velocity estimates if we strung the velocity information inside the model. For this reason, we ignore limb velocities, and use configuration of the body parts as our only feature. However, one should note that velocity and acceleration is a useful clue when differentiating run and walk motions.

In this section, we first describe how we form our base units on acts and actions, then how we string them together to form complex activity models.

### 5.1.1 Acts in short timescales

Individual frames are a poor guide to what the body is up to, not least because transduction is quite noisy and the frame rate is relatively high (15-30Hz). We expect better behaviour from short runs of frames. At the short timescale, we represent motion with three frame long snippets of the lifted 3D representation. We form one snippet for each leg and one for each arm; we omit the torso, because torso motions appear not to be particularly informative in practice (see section 6). Each limb in each frame is represented with the vector quantized value of the snippet centered on that frame. That is, we apply k-means to the 3D representation of snippets the limbs. We use 40 as the number of clusters in vector quantization, for each limb. One can utilize different levels of quantization, but our experiments show that for this dataset, using 40 for each limb provides good enough generalization.

### 5.1.2 Limb action models

Using a vague analogy with speech, we wish to build a large dynamical model with the minimum of parameter estimation. In speech studies, in order to recognize words, phoneme models are built and joined together to form word models [72]. By learning phoneme models and joining them together, word models share information within the phoneme framework, and this makes building large vocabularies of word models possible.

By using this analogy, we first build a model of the action of each limb (arms, legs) for a range of actions, using Hidden Markov Models (HMM's [73]) that emit vector quantized snippets we formed in the previous step. We choose a set of 9 actions by hand, with the intention of modelling our motion capture collection reasonably well; the collection is the research collection of motion capture data released by Electronic Arts in 2002, and consists of assorted football movements. Motion sequences from this collection are sorted into actions using the labelling of [7]. The original annotation includes 13 action labels; we have excluded actions with the direction information (3 actions named `turn left, turn right, backwards`) and observed that `reach` and `catch` actions do not differ significantly in practice, so we joined the

data for these two actions and labelled them as `reach` altogether. Moreover, this labelling is adapted to have separate action marks for each limb. Since actions like `wave` cannot be definable for legs, we only used a subset of 6 actions for labelling legs and 9 for labelling arms.

For each action, we fit to the examples using maximum likelihood, and searching over 3-10 state HMM models. Experimentation with the structures shows that 3-state models represent the data well enough. Thus, we take 3-state HMMs as our smallest unit for action representation. Again, we emphasize that the action dynamics are completely built on 3D motion capture data.

### 5.1.3 Limb activity models

Having built atomic action models, we now string the limb models into a larger HMM by linking states that have similar emission probabilities. That is, we put a link between states $m$ and $n$ of the different action models $A$ and $B$ if the distance

$$dist(A_m, B_n) = \sum_{o_m=1}^{N} \sum_{o_n=1}^{N} p(o_m)p(o_n)C(o_m, o_n) \tag{5.1}$$

is minimal. Here, $o_m$ and $o_n$ are the emissions, $p(o_m)$ and $p(o_n)$ are the emission probabilities of respective action model states $A_m$ and $B_n$, $N$ is the number of possible emissions and $C(o_m, o_n)$ is the Euclidean distance between the emissions centers, which are the cluster centers of the vector-quantized 3D joint points.

The result of this linkage is a dynamical model for each limb that has a rich variety of states, but is relatively easily learned. States in this model are grouped by limb model, and we call a group of states corresponding to a particular limb model a **limb activity model** (Figure 5.2). While linking these states, we assign uniform probability to transition between actions and transition to the same action. That is, the probability of the action staying the same is set equal to the probability of transferring to another action. This is analogous to joining phoneme models to recognize words in speech recognition. This is loosely a generative model, we compute the probability that each sequence is generated by a certain set of action HMMs.

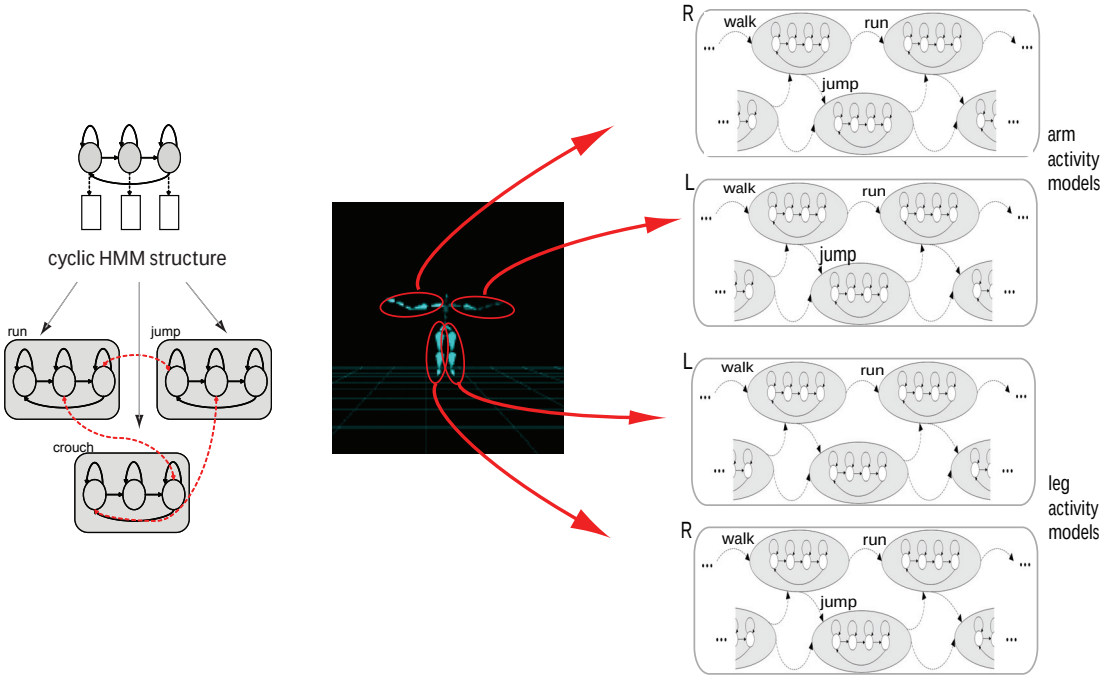



Figure 5.2: Formation of activity models for each of the body parts. First, single action HMMs for left leg, right leg, left arm, right arm are formed using motion capture dataset. Second, single action HMMs are joint together by linking the states that have similar emission probabilities.

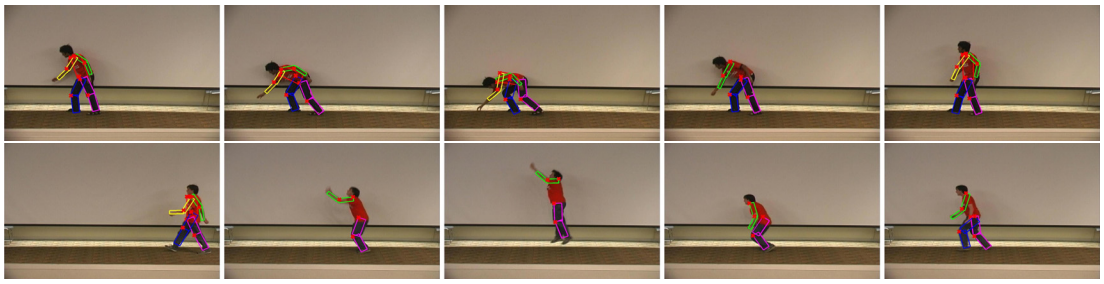


Figure 5.3: Here are some example good tracks for the UIUC video dataset. These are two sequences performed by two different actors wearing different outfits. **Top:** `stand-pickup` sequence. **Bottom:** `walk-jump-reach-walk` sequence.

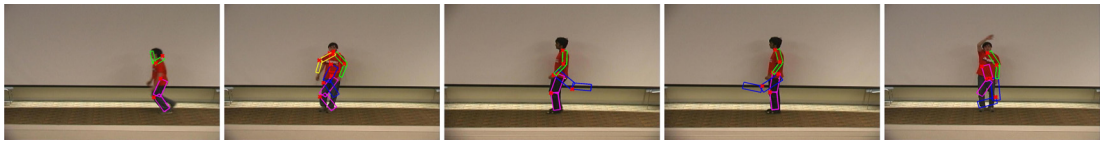


Figure 5.4: Due to motion blur and similarities in appearance, some frames are out of track. **first:** appearance and motion blur error **second:** legs mixed up because of rectangle search failure on legs. **third and fourth:** one leg is occluded by the other leg, the tracker tries to find second leg, mistaken by the solid dark line **fifth:** motion blur causes tracker to miss the waving arm, legs scrambled.

## 5.2 Transducing the body

### 5.2.1 Tracking

We track motion sequences with the tracker of Ramanan *et al.* [76]; this tracker obtains an appearance model by detecting a lateral walk pose, then detects instances in each frame using the pictorial structure method of [29]. The advantage of using this tracker is that it is highly robust to occlusions and complex backgrounds. There is no need for background modelling, and this tracker has been shown to perform well on changing backgrounds (see also section 6.5). Moreover, it is capable of identifying the distinct limbs, which we need to form our separate limb action models.

Example outputs of the tracker is given in Fig 5.3 and Fig 5.4. In Fig 5.3, the tracker is able to spot most of the body parts in these sequences. However, in most of the sequences, especially in lateral views, only two out of four limbs are tracked

because of the self-occlusions.

Kinematic tracking is known to be hard (see the review in [33]) and, while the tracker is usable, it has some pronounced eccentricities (Figure 5.4 [77]). Note that all such bad tracks are a part of our test collection and non-perfect tracking introduces considerable amount of noise to our motion understanding procedure. By lifting 2D tracks to 3D, we want to suppress the effects of such noise as much as possible.

### 5.2.2 Lifting 2D tracks to 3D

The tracker reports a 2D configuration of a puppet figure in the image (Figure 5.3), but we require 3D information. Several authors have successfully obtained 3D reconstructions by matching projected motion capture data to image data by matching **snippets** of multiple motion frames [41, 42, 75]. A complete sequence incurs a per-frame cost of matching the snippet centered at the frame, and a frame-frame transition cost which reflects (a) the extent of the movement and (b) the extent of camera motion. The best sequence is obtained with dynamic programming. The smoothing effect of matching snippets — rather than frames — appears to significantly reduce reconstruction ambiguity (see also the review in [33]).

The disadvantage of the method is that one may not have motion capture that matches the image well, particularly if one has a rich collection of activities to deal with. We use a variant of the method. In particular, we decompose the body into four quarters (two arms, two legs). We then match the legs using the snippet method, but allowing the left and right legs to come from different snippets of motion capture, making a search over 20 camera viewing directions. The per-frame cost must now also reflect the difference in camera position in the root coordinate system of the motion capture; for simplicity, we follow [75] in assuming an orthographic camera with a vertical image plane. We choose arms in a similar manner conditioned on the choice of legs, requiring the camera to be close to the camera of the legs. In practice, this method is able to obtain lifts to quite rich sequences of motion from a relatively small motion capture collection. Our lifting algorithm is given in Algorithm 1.

**Algorithm 1** Lifting 2d tracks to 3d

**for** each camera $c \in C$ **do**
  **for all** pose $p \in mocap$ **do**
    $\sigma_{pc} \leftarrow projection(p, c)$
  **end for**
  $camera\_transition\_cost\ \delta(c_i, c_j) \leftarrow (c_i - c_j) \times \alpha$
**end for**
**for** each $l_t \in L$ (leg segments in 2D) **do**
  **for all** $p \in mocap$ and $c \in C$ **do**
    $\lambda(l_t, \sigma_{pc}) \leftarrow match\_cost(\sigma_{pc}, l_t)$
    $\gamma(l_t, l_{t+w}) \leftarrow transition\_cost(\lambda(l_t, \sigma_{pc}), \lambda(l_{t+w}, \sigma_{pc}))$
  **end for**
**end for**
do dynamic programming over $\delta, \lambda, \gamma$ for $L$
$c_{legs} \leftarrow$ (minimum cost camera sequence)
**for** each $a_t \in A$ (arm segments in 2D) **do**
  **for** $c_\epsilon \leftarrow$ neighborhood $\epsilon$ of $c_{legs}$ and pose $p \in mocap$ **do**
    compute $\lambda(a_t, \sigma) \leftarrow match\_cost(\sigma_{pc_\epsilon}, a_t)$
    compute $\gamma(a_t, a_{t+w}) \leftarrow transition\_cost(\lambda(a_t, \sigma_{pc_\epsilon}), \lambda(a_{t+w}, \sigma_{pc_\epsilon}))$
  **end for**
**end for**
do dynamic programming over $\delta, \lambda, \gamma$ for $A$

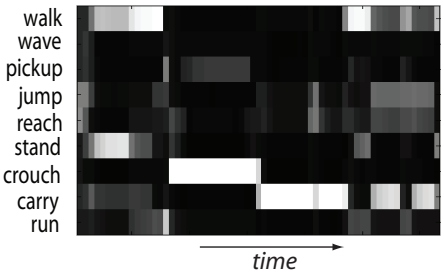



Figure 5.5: Posterior probability map of a `walk-pickup-carry` video of an arm. This probability map corresponds to a run of forward algorithm through the activity HMM for this particular video.

### 5.2.3 Representing the body

We can now represent the body's behaviour for any sequence of frames with $P(\text{limb activity model}|\text{frames})$. The model has been built entirely on motion capture data. By computing a forward-algorithm pass of the lifted sequences over the activity models, we get a posterior probability map representation for each video, which indicates the likelihood of each snippet to be in a particular state of the activity HMMs over the temporal domain.

**Hidden Markov Models(HMMs)** are statistical models that aim to describe the sequence of observations by means of discovering the hidden states from the observable parameters. HMMs obey the Markovian property, where the current state depends on the previous state(s) and are extensions of Markov chains. In Markov chains, the state outputs are deterministic and each state corresponds to a single (observable) event [72]. On the other hand, in hidden Markov models, the observations are probabilistic functions of the states, and the states cannot be directly deduced from the sequence of observations (i.e. they are hidden).

Each HMM is defined by two sets of stochastic processes; first one describes the state transition probabilities and second is the stochastic process that generate the observations for each state [61]. The task is to estimate these two processes that best explain the training data.

Formally, HMMs are identified with five main components, which are (following [72])

- $N$, the number of states
- $M$, the number of possible observations per state, or the alphabet size, where the individual symbols are denoted as $V = \{v_1, v_2, \ldots, v_M\}$
- $A = \{a_{ij}\}$, the state transition probability distribution where

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N. \tag{5.2}$$

- $B = \{b_j(k)\}$, the observation probability distribution where

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j], \quad \begin{array}{r} 1 \leq j \leq N, \\ 1 \leq k \leq M. \end{array} \tag{5.3}$$

- $\pi_i$ , the initial state distribution where

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N. \tag{5.4}$$

There are three fundamental problems of Hidden Markov Models. These are evaluation, decoding and learning problem. The evaluation problem refers to the case where, given the model, we compute how likely that a particular output sequence is generated by that HMM model. This problem is solved by using forward algorithm, as described in [72]. Decoding problem tries to find the sequence of states that most likely generated the given particular output, and is solved by the viterbi algorithm. The learning (i.e training) problem, on the other hand, tries to find the underlying stochastic processes (the model parameters $A, B, \pi$) that maximizes the fitting to the training data. The learning problem is solved by Baum-Welch method (Expectation Maximization (EM)). In our case, we first solve the learning problem and estimate the parameters of the action HMMs by using EM algorithm. Then, the sequences are decoded (as a case of decoding problem) using forward algorithm to get the possible pathway of observations through the action states.

More formally, the posterior probability of a set of action states $\lambda = (s_1, \ldots, s_t)$ given a sequence of observations $\sigma_k = o_1, o_2, \ldots, o_t$ and model parameters $\theta$ can be computed from the joint. In particular, note

$$\begin{aligned} P(\lambda | \sigma_k, \theta) &\propto P(\lambda, \sigma_k | \theta) \\ &= P(s_1) \left( \prod_{j=1}^{t-1} P(o_j | s_j) P(s_{j+1} | s_j) \right) P(o_t | s_t) \end{aligned} \tag{5.5}$$

where the constant of proportionality $P(\sigma_k)$ can be computed easily with the forward-backward algorithm (for more details, see, for example [72]). We follow convention and define the forward variable $\alpha_t(i) = P(q_t = i, o_1, o_2, \ldots, o_T|\theta)$, where $q_t$ is the state of the HMM at time $t$ and $T$ is the total number of observations. Similarly, the backward variable $\beta_t(i) = P(o_{t+1}, \ldots, o_T|q_t = i, \theta)$. We write $b_j(o_t) = P(o_t|q_t = j)$, $a_{ij} = P(q_t = j|q_{t-1} = i)$ and so have the recursions

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \left[ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] \tag{5.6}$$

$$\beta_t(j) = \sum_{i=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \tag{5.7}$$

and

$$\alpha_1(i) = \pi_i b_i(o_1) \tag{5.8}$$

$$\beta_T(i) = 1 \tag{5.9}$$

This gives

$$P(\sigma_k) = \sum_{i=1}^{N} \alpha_T(i) \tag{5.10}$$

Our activity model groups states with an equivalence relation. For example, several different particular configurations of the leg might correspond to walking. We can compute a posterior over these groups of states in a straightforward way. We assume we have a total of $M \leq N$ groups of states. Now assume we wish to evaluate the posterior probability of a sequence of state groups $\lambda_g = (g_1, \ldots, g_t)$ conditioned on a sequence of observations $\sigma_k = (o_1, \ldots, o_t)$. We can regard a sequence of state groups as a set of strings $\Lambda_g$, where a string $\lambda \in \Lambda_g$ if and only if $s_1 \in g_1$, $s_2 \in g_2$, ..., $s_t \in g_t$. Then we have

$$P(\lambda_g, \sigma_k) = \sum_{\lambda \in \Lambda_g} P(\lambda, \sigma_k) \tag{5.11}$$

This allows us to evaluate the posterior on activity models (see, for example, Figure 5.5).

Figure 5.5 shows an example posterior probability map. Here, the rows represent the atomic action models and columns represent the time dimension. By examining this probability map, one can infer the timeline of the possible changes inside the

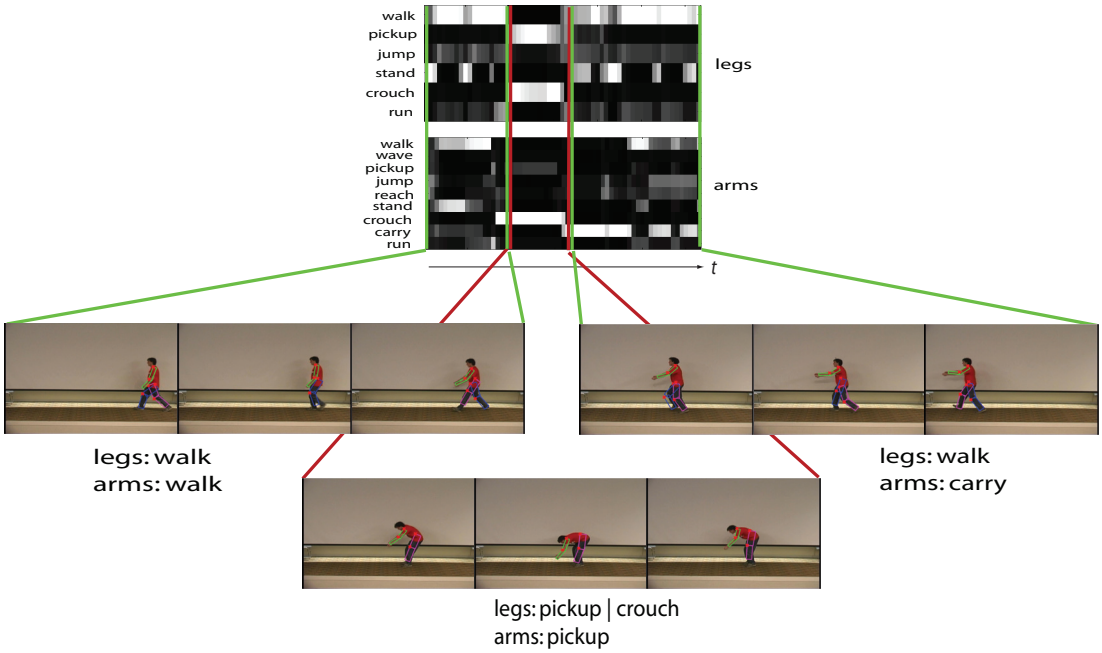



Figure 5.6: An example query result of our system. **Top:** Average HMM posteriors for the legs and arms of sequence `walk-pickup-carry` (performed by a male subject) are shown. As it can be seen, maximum likelihood goes from one action HMM to the other within the activity HMM as the action in the video changes. This way, we achieve automatic segmentation of activities and there is no need to use other motion segmentation procedures. **Bottom:** Corresponding frames from the subsequences are shown. This sequence is correctly labeled and segmented as `walk-pickup-carry` as the corresponding query is evaluated. Using activity models for each body part, we compute posteriors of sequences. After that, HMM posteriors for right and left parts of the body are queried together using finite state automata of the query string.

activity by following the change of intensities through the rows, i.e. action models. The action models that constitute up the activity models are quite discriminative, therefore we can expect a good search for a composition. Moreover, the action models give a good segmentation in and of themselves. Despite some noise, we can clearly observe transitions between different actions within the video.

As example sequences in Figure 5.6 and 5.7 indicate, this representation is quite competent at discriminating between different labellings for motion capture data. In addition, we achieve automatic segmentation of activities using this representation. There is no need for explicit motion segmentation, since transitions between action HMM models simply provide this information.

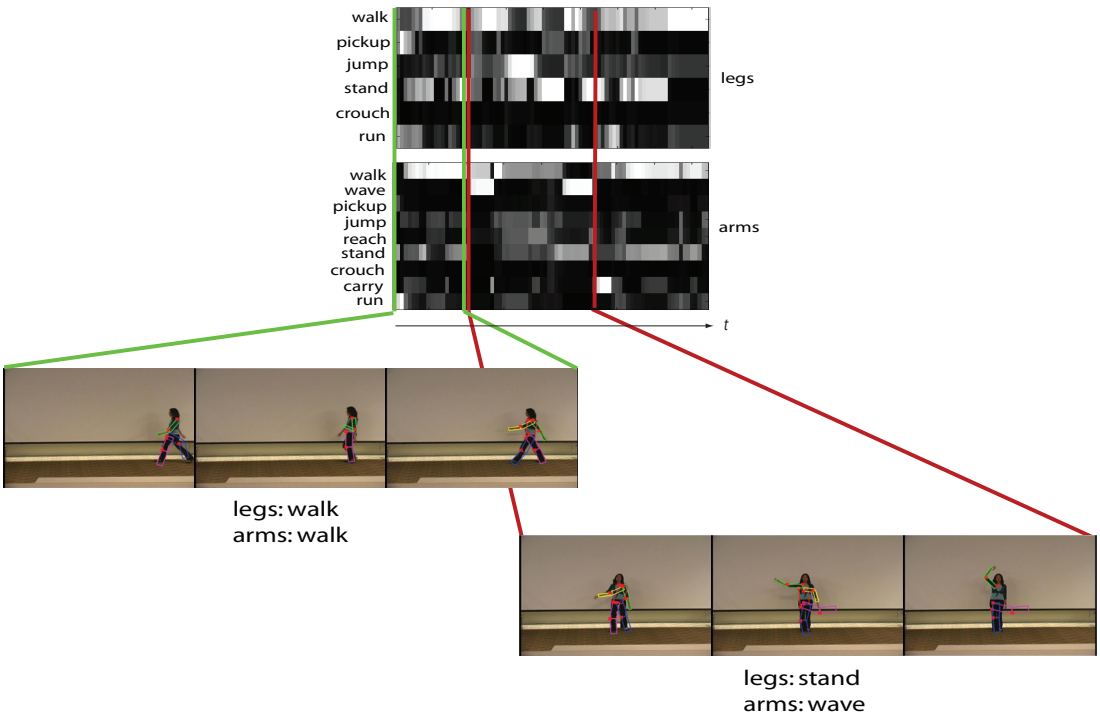



Figure 5.7: Another example sequence from our system, performed by a female subject. In this sequence, the subject first walks into the scene, stops and waves for some time, and then walks out of the sequence. A query for `walk-wave-walk` for arms and `walk-stand-walk` for legs returned this sequence as top one, despite the noise in tracking. Again, by examining the posterior maps for each limb, we can identify the transitions between actions.**Top:** Posterior probability maps for legs and arms. **Bottom:** Corresponding frames from the correctly identified subsequences.

## 5.3 Querying for Activities

We can compute a representation of what the body is doing from a sequence of video. By using this representation, we would like to be able to build complex queries of composite activities, such as carrying while standing, or waving while running. We can address composition across the body because we can represent different limbs doing different things; and composition in time is straightforward with our representation.

This suggests thinking of querying as looking for strings, where the alphabet is a product of possible activities at limbs and locations in the string represent locations in time. Generally, we do not wish to be precise about the temporal location of particular activities, but would rather find sequences where there is strong evidence for one activity, followed by strong evidence for another, and with a little noise scattered about. In turn, it is natural to start by using regular expressions for motion queries (we see no need for a more expressive string model at this point).

An advantage of using regular expressions is that it is relatively straightforward to compute

$$\sum_{\text{strings matching RE}} P(\text{string}|\text{frames}) \tag{5.12}$$

which we do by reducing the regular expression to a finite state automaton and then computing the probability this automaton reaches its final state using a straightforward sum-product algorithm.

This query language is very simple: Suppose we want to find videos where the subject is walking and waving his arms at the same time. For legs, we form a walk automaton. For arms, we form a wave automaton. We simultaneously query both limbs with these automata. Figures 5.9 and 5.10 show the corresponding automata for example queries.

**Finite State Representation for Activity Queries:** A finite state automaton (FSA) is defined with the quintuple $(Q, \Sigma, \delta, s_0, F)$, where $Q$ is the finite non-empty set of states of the FSA, $\Sigma$ is the input alphabet, $\delta$ is the state transition function where $\delta : Q \times \Sigma \rightarrow Q$, $s_0$ is the (set) of initial states, and $F$ is the set of final states. In our

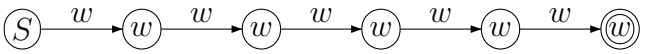



Figure 5.8: The FSA for a single action is constructed based on its unit length. Here, the expansion of the `walk` FSA is shown (w represents walk). As an example, unit length of `walk` is set to 5 frames ($u_w = 5$). So the corresponding FSA consist of five states and the probability of it reaching its final state requires that we observe five consecutive frames of `walk`.

representation, each state $q_i \in Q$ corresponds to the case where the subject is inside a particular action. Transitions between states ($\delta$) represent the actions taking place. Transitions of the form $x^{u_x}$ means action $x$ sustained for $u_x$ length, which means that actions shorter than their specified unit length do not cause the FSA to change its state. More specifically, each $x^{u_x}$ (shown over the transition arrows) represents a smaller FSA on its own, as shown in Figure 5.8. This small FSA reaches in its end state when the action is sustained for $u_x$ number of frames. This regulation is needed in order to eliminate the effect of short-term noise.

While forming the finite state automata, as in Figure 5.9, each action is considered to have a unit length $u_x$. A query string is converted to a regular expression, and then to an FSA based on these unit lengths of actions. Unit action length is based on two factors: first is the fps of the video, second is the action's level of sustainability. Actions like walking and running are sustainable; thus their unit length is chosen to be longer than those of localizable actions, like jump and reach.

We have an FSA $F$, and wish to compute the posterior probability of any string accepted by this FSA, conditioned on the observations. We write $\Sigma_F$ for the strings accepted by the FSA. We identify the alphabet of the FSA with states — or groups of states — of our model, and get

$$P(F|o_1, \ldots, o_T, \theta) \propto \sum_{\sigma \in \Sigma_F} P(\sigma, o_1, \ldots, o_T|\theta) \tag{5.13}$$

where the constant of proportionality can be obtained from the forward- backward algorithm, as in section 5.2.3. The term $\sum_{\sigma \in \Sigma_F} P(\sigma, o_1, \ldots, o_T|\theta)$ requires some care.

We label the states in the FSA with indices $1, ..Q$. We can compute this sum with a recursion in a straightforward way: Write

$$\begin{aligned} Q_{ijs} &= P\{\text{a string of length } i \text{ that takes } F \text{ to state } j \text{ and has last element } s, \text{ joint with } o_1, ..., o_i\} \\ &= \sum_{\sigma \in \text{strings of length } i \text{ with last character } s \text{ that take } F \text{ to } j} P(\sigma, o_1, \ldots, o_i|\theta) \end{aligned} \tag{5.14}$$

Write Pa$(j)$ for the parents of state $j$ in the FSA (that is, the set of all states such that a single transition can take the FSA to state $j$). Write $\delta_{i,s}(j) = 1$ if $F$ will transition from state $i$ to state $j$ on receiving $s$ and zero otherwise; then we have

$$Q_{1js} = \sum_{u \in s_0} P(s, o_1|\theta)\delta_{u,s}(j) \tag{5.15}$$

and

$$Q_{ijs} = \sum_{k \in \text{Pa}(j)} \delta_{k,s}(j) P(o_i|s, \theta) \left[ \sum_{u \in \Sigma} P(s|u, \theta) Q_{(i-1)ku} \right] \tag{5.16}$$

Then

$$\sum_{\sigma \in \Sigma_F} P(\sigma, o_1, \ldots, o_T|\theta) = \sum_{u \in \Sigma, v \in s_e} Q_{Tvu} \tag{5.17}$$

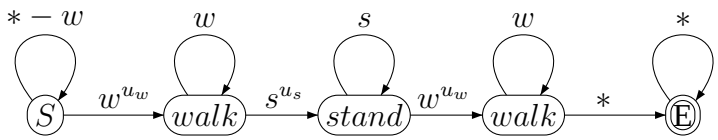
and we can evaluate $Q$ using the recursion. Notice that nothing major changes if each item $u$ of the FSA's alphabet represents a set of HMM states (as long as the sets form an equivalence relation). We must now modify each expression to sum states over the relevant group. So, for example, if we write $s_u$ for the *set* of states represented by the alphabet term $u$, we have

$$Q_{1ju} = \sum_{u \in s_0} \sum_{v \in s_u} P(v, o_1|\theta)\delta_{u,s}(j) \tag{5.18}$$

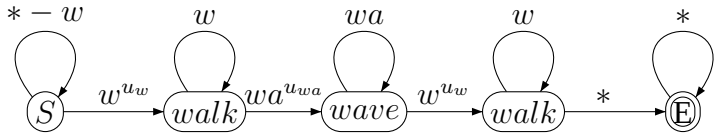
and

$$Q_{iju} = \sum_{k \in Pa(j), v \in s_u} \delta_{k,v}(j) P(o_i|v, \theta) \left[ \sum_{u \in \Sigma, w \in s_u} P(v|w, \theta) Q_{(i-1)ku} \right] \tag{5.19}$$

A tremendous attraction of this approach is that no visual example of a motion is required to query; once one has grasped the semantics of the query language, it is easy to write very complex queries which are relatively successful.




(a)




(b)

Figure 5.9: To retrieve complex composite activities, we write separate queries for each of the body parts. Here, example query FSAs for a sequence where the subject walks into the view, stops and waves and then walks out of the view are shown. **Top:** FSA formed for the legs `walk-stand-walk`. **Bottom:** The corresponding query FSA for the arms with the string `walk-wave-walk`. Here, $w$ is for walk, $s$ for stand, $wa$ for wave and $u_x$'s are the corresponding unit lengths for each action $x$.

The alphabet from which queries are formed consists in principle of $6^2 \times 9^2$ terms (one has one choice each for each leg and each arm). We have found that the tracker is not sufficiently reliable to give sensible representations of both legs (resp. arms). It is often the case that one leg is tracked well and the other poorly, mainly because of the occlusions. We therefore do not attempt to distinguish between legs (resp. arms), and reduce the alphabet to terms where either leg (resp. either arm) is performing an action; this gives an alphabet of $6 \times 9$ terms (one choice at the leg and one at the arm). This is like a noisy OR operation over the signals coming from top and bottom parts of the body. When any of the signals are present we take the union of them to represent the body pose.

Using this alphabet, we can write complex composite queries, for example, searching for strings that have several `(l-walk; a-walk)`'s followed by several `(l-stand; a-wave)` followed by several `(l-walk; a-walk)` yields sequences where a person walks into view, stands and waves, then walks out of view (see Figure 5.9 for corresponding FSAs). Figure 5.10 demonstrates an example query for

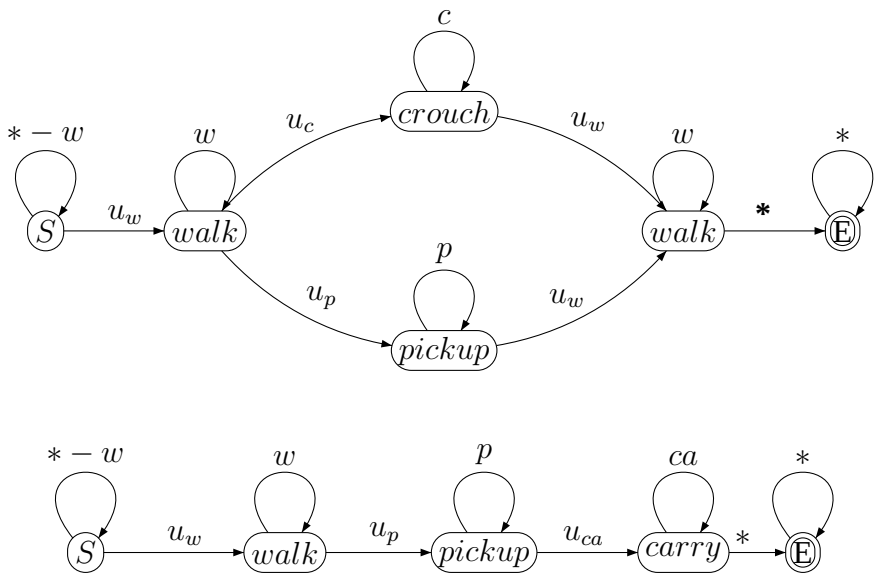



Figure 5.10: Query FSA for a video where the person walks, picks something up and carries it. Query for a video where the person walks, pickups something and carries it. Here, $w$ is for walk, $c$ for crouch, $p$ for pickup and $ca$ is for carry actions. Notice the different and complex representation achievable by writing queries in this form. Arms and legs are queried separately, composited across time and body. Also note that, since `pickup` and `crouch` actions are very similar in dynamics for the legs, we can form an `OR` query and do more wide-scale searches.

`walk-pickup-carry` activity sequence. Notice the different and complex representation achievable by writing queries in this form. Arms and legs are queried separately, composited across time and body. Also note that, since `pickup` and `crouch` actions are very similar in dynamics for the legs, we can form an `OR` query and do more wide-scale searches as in Fig 5.10.

# Chapter 6

# Experiments on Complex Human Activities

This chapter presents our experimental evaluation of using limb activity models for complex activity recognition. The experimental settings and evaluation datasets for complex activity recognition is considerably different than the single action recognition case. Here, we need to access the distinct body parts, for this reason, we used the tracker of Ramanan *et al.*, which is equipped with explicitly identifying the body parts.

## 6.1 Experimental Setup

In this section, we describe the setup and datasets of our experiments. The nature of the human activities inside these video sequences are more complex; the actions are composed across time and across body. In addition, there is a considerable amount of variation in the clothings, viewpoint and background. We test quite different cases and experiment with a wide range of queries.

Table 6.1: Our collection of video sequences, named by the instructions given to actors.

| Context | # videos | Context | # videos |
|---|---|---|---|
| crouch-run | 2 | run-backwards-wave | 2 |
| jump-jack | 2 | run-jump-reach | 5 |
| run-carry | 2 | run-pickup-run | 5 |
| run-jump | 2 | walk-jump-carry | 2 |
| run-wave | 2 | walk-jump-walk | 2 |
| stand-pickup | 5 | walk-pickup-walk | 2 |
| stand-reach | 5 | walk-stand-wave-walk | 5 |
| stand-wave | 2 | crouch-jump-run | 3 |
| walk-carry | 2 | walk-crouch-walk | 3 |
| walk-run | 3 | walk-pickup-carry | 3 |
| run-stand-run | 3 | walk-jump-reach-walk | 3 |
| run-backwards | 2 | walk-stand-run | 3 |
| walk-stand-walk | 3 | | |

## 6.1.1 Datasets

### 6.1.1.1 UIUC Complex Activity Dataset

We collected our own set of motions, involving three subjects wearing a total of five different outfits in a total of 73 movies (15Hz). Each video shows a subject instructed to produce a complex activity. The sequences differ in length. The complete list of activities collected is given in Table 6.1 and the example frames for this dataset is given in Fig 6.1.

### 6.1.1.2 Viewpoint Dataset

For viewpoint evaluation, we collected videos of 5 actions: jog, jump, jumpjack, wave and reach. Each action is performed in 8 different directions to the camera, making a total dataset of 40 videos (30Hz). Figure 6.2 shows example frames of this dataset.

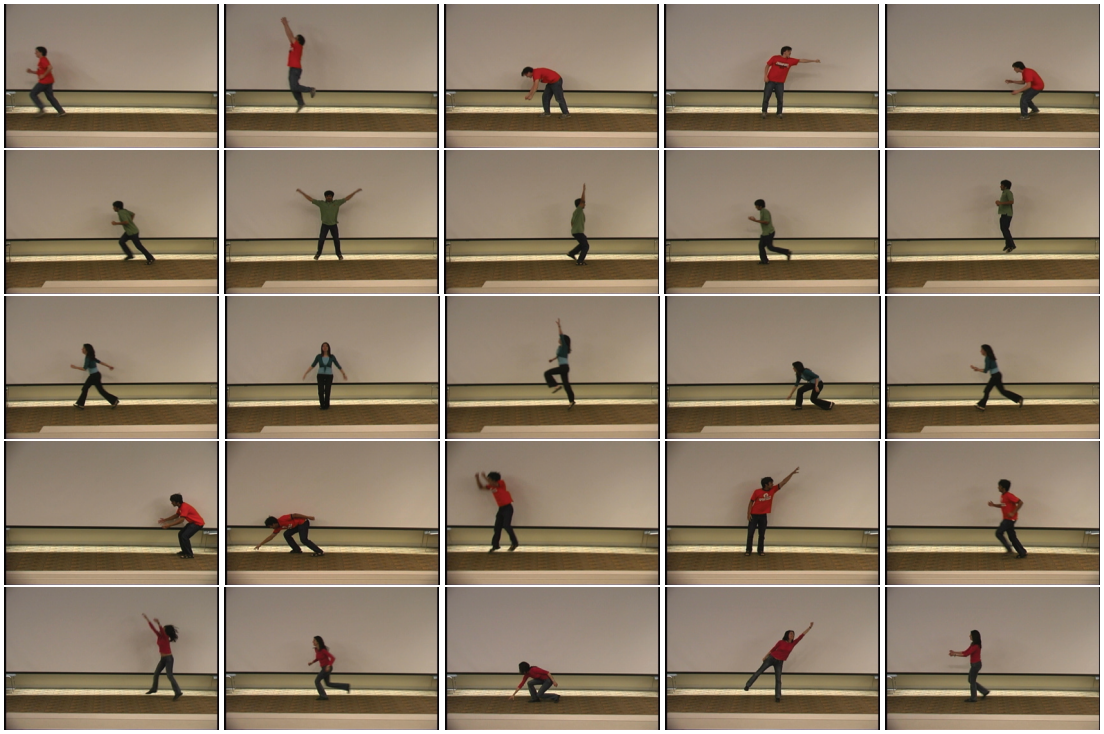


Figure 6.1: Example frames from UIUC complex activity dataset. Here, there are three actors wearing five different outfits, performing 13 different combinations of activities. The resulting dataset has 73 movies in total.

Figure 6.2: Example frames from our dataset of single activities with different views. **Top row:** Jogging 0 degrees, Jump 45 degrees, jumpjack 90 degrees, reach 135 degrees. **Bottom row:** wave 180 degrees, jog 225 degrees, jump 270 degrees, jumpjack 315 degrees.

Figure 6.3: Example frames from the Friends dataset, which consists of 19 short sequences compiled from the Friends TV series (from Episode 9 of Season 3), where the actors play football in the park. This is a challenging dataset, in which there are lots of camera movement, scale and orientation changes, zoom-in and out effects. Occlusions make the tracking even harder in this dataset.

#### 6.1.1.3 Friends Dataset

For evaluating our system on complex backgrounds and also on football movements, we used video shootage from the TV series Friends. We have extracted 19 sequences of varying activities from the episode in which the characters play football in the park. The result is an extremely challenging dataset; the characters change orientation frequently, the camera moves, there are zoom-in and zoom-out effects and a complex and changing background. Different scales and occlusions make tracking even harder. In Figure 6.3, we show example frames from this dataset with superimposed tracks.

### 6.1.2 Evaluation Method

We evaluate the performance of our system over a set of queries, using mean average precision (MAP) of the queries. Average precision of a query is defined as the area under the precision-recall curve for that query and a higher average precision value

means that more relevant items are returned earlier.

More formally, average precision $AveP$ over a set $S$ is defined as

$$AveP = \frac{\sum_{r=1}^{N}(P(r) \times rel(r))}{\text{number of relevant documents in S}}$$

Here, $r$ is the rank of the item, $N$ is the number of retrieved items and $rel(r)$ is the binary relevance vector for each item in $S$ and $P(r)$ precision at a given rank.

**Clothing** presents a variety of problems. We know of no methods that behave well in the presence of long coats, puffy jackets or of skirts. Our subjects wear a standard uniform of shirt and trousers. However, as figure 6.7 shows, the colour, arm-length and looseness of the shirts varies, as does the cut of the trousers and the presence of accessories (a jersey). These variations are a fairly rich subset of those that preserve the silhouette. Our method is robust to these variations, and we expect it to be robust to any silhouette preserving change of clothing.

**Controls:** In order to analyse the performance of our approach, we implemented three controls. Control 1 is single action SVM classifiers over raw 2D tracks (details in section 6.3.2.1). We expect that discriminative methods applied to 2D data perform poorly because intra-class variance overwhelms available training data. In comparison, our method benefits by being able to estimate dynamical models on motion capture dataset. Control 2 is action SVMs built on 3D lifts of the 2D tracks (for details see section 6.3.2.2). Although they have view-invariance aspect, we also expect them performing poorly, because they suffer from data shortage and noise in lifts. And finally, Control 3 is the SVM classifiers over 3D motion capture dataset (details in section 6.3.2.3). They are also insufficient in tolerating the different levels of sustainability and different speeds of activities. This also causes problems with the composition. On contrary, our model supports high level of composition and its generative nature handles different lengths of activities easily.

## 6.2 Expressiveness of Limb Activity Models

**Limb activity models** were fit using a collection of 10938 frames of motion capture

data released by Electronic Arts in 2002, consisting of assorted football movements. To model our motion capture collection reasonably well, we choose a set of 9 actions. While these actions are abstract building blocks, the leg models correspond reasonably well to: run, walk, stand, crouch, jump, pickup (total of 6 actions). Similarly, the arm models correspond reasonably well to: run, walk, stand, reach, crouch, carry, wave, pickup, jump motions (total of 9 actions).

Local dynamics is quite a good guide to a motion in the motion capture data set. Figure 6.4 shows HMM interpretation of these dynamics. The posterior for each model applied to labelled motion capture data is given. These images represent the expressive and generative power of each action HMM. For example, `pickup` HMM for the legs gives high likelihood for `pickup` and `crouch` action, whereas `crouch` HMM for the legs is more certain when it observes a `crouch` action, therefore it produces a higher posterior as opposed to `pickup`. The asymmetry present in this figure is due to the varying number of training examples available in motion capture dataset for each action. The higher the number of examples for an action, the better HMMs are fit. This can be interpreted as a class confusion matrix within the motion capture dataset itself. Most of the confusion occurs between dynamically similar actions. For example, for `pickup` motion, the leg HMMs may fire `pickup` or `crouch` motions. These two actions are in fact very similar in dynamics. Likewise, for `reach` motion, arm HMMs show higher posteriors for `reach`, `wave` or `jump` motions.

Limb activity models require that 3D coordinates of limbs to be vector quantized. The choice of procedure has some effect on the outcome and details of this procedure is explored in Section 6.2.1.

We expect these HMM's to simulate rendered activity extremely poorly, as they are not constructed to produce good transitions between frames. We are not claiming that the generative model concentrates probabilities only on correct human actions, and we don't believe that any other work in activity makes this claim; the standards of performance required to do credible human animation are now extremely high (eg [52, 54, 8]; review in [33]), and it is known to be very difficult to distinguish automatically between good and bad animations of humans [78, 44, 33]. Instead, we believe that the probability that appears on actions that are not natural, does not present difficulties as

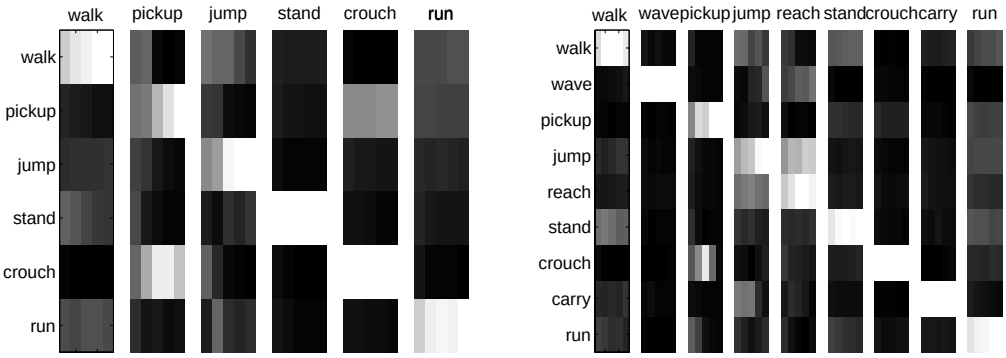



Figure 6.4: Each column represents 5 frame average HMM posteriors for the motion capture sequences (**left:**legs **right:**arms). This image can also be interpreted as a confusion matrix between actions.

long as the models are used for inference, and our experimental evidence bears this out. Crucially, when one infers activity labels from video, one can avoid dealing with sequences that do not contain natural human motion.

### 6.2.1 Vector Quantization for Action Dynamics

We vector quantize 3D coordinates of the limbs when forming the action models. This quantization step is useful to have a more general representation of the domain. We use k-means as our quantization method. Since k-means is very dependent on the initial cluster centers, we run each clustering 10 times and choose the best clusters such that the inter-cluster distance is maximized and intra-cluster distance is minimized. Our experiments show that when we choose number of clusters $k$ in k-means as low as 10, the retrieval process suffers from information loss due to excessive generalization. Using $k = 40$ gives the best results over this dataset. Note that, one can try different levels of quantization for different limbs, however, our empirical evaluation shows that doing so does not provide a significant performance improvement. Figure 6.5 shows the effect of choosing the number of clusters.

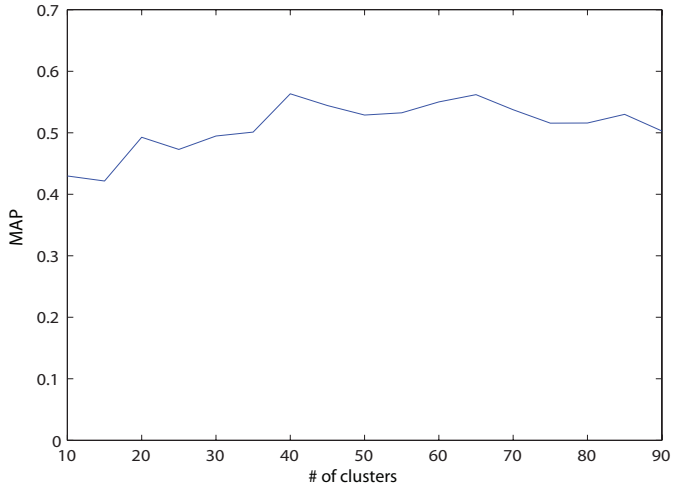



Figure 6.5: The result of choosing $k$ in k-means. We have observed that this choice affects the ranking considerably. More clusters force sequences which look more like the training set to have a higher rank. However, the motions that are less alike in appearance, but similar in semantics are penalized.

## 6.3 Searching

We evaluate our system by first identifying an activity to search for, then marking relevant videos, then writing a regular expression, and finally determining the recall and precision of the results ranked by $P(\text{FSA in end state}|\text{sequence})$. On the traditional simple queries (`walk, run, stand`), MAP value is 0.9365; only a short sequence of `run` action is confused with `walk` action. Figure 6.7 and Figure 6.8 show search results for more complex queries. We have used $k$=40 in vector quantization. Our method is able to respond to complex queries quite effectively. The biggest difficulty we faced was to find an accurate track for each limb due to the discontinuity in track paths and left/right ambiguity of the limbs. That's why some sequences are identified poorly.

We have evaluated several different types of search. In Type I queries, we encoded activities where legs and arms are doing different actions simultaneously, for instance "walking while carrying". In Type II queries, we evaluated the cases where there are two consecutive actions, same for legs and arms (like a `crouch` followed by a `run`). Type III queries search for activities that are more complex; these are activities

Table 6.2: The Mean Average Precison(MAP) values for different types of queries. We have three types of query here. Type I: single activities where there is a different action for legs and arms (ex: walk-carry). Type II: two consecutive actions like `crouch` followed by a `run`. Type III: activities that are more complex, consisting of three consecutive actions where different body parts may be doing different things (ex: `walk-stand-walk` for legs; `walk-wave-walk` for arms.

| Query type | MAP |
|---|---|
| Type I | 0.5562 |
| Type II | 0.5377 |
| Type III | 0.5902 |

involving three consecutive actions where different limbs may be doing different things (ex: `walk-stand-walk` for legs; `walk-wave-walk` for arms). MAP value for these sets of complex queries is 0.5636 with our method.

The performance over individual type of activities is presented in Table 6.2. Based on this evaluation, we can say that our system is more successful in retrieving complex activities as in Type III queries. That's mostly because complex activities occur within longer sequences which are less affected by the short-term noise of tracking and lifting.

### 6.3.1 Torso exclusion

In our method, we omit the torso information and query over the limbs only. This is because we found that torso information is not particularly useful. The results demonstrating this case is given in Figure 6.6. When we query using the whole body, including torso, we get an Mean Average Precision of 0.501, whereas if we query using limbs only, we get a MAP of 0.5636. We conclude that using torso is not particularly informative. This is mostly because in our set of actions, the torso HMMs fire high posteriors for more than one action, and therefore, they don't help much in discriminating between actions.

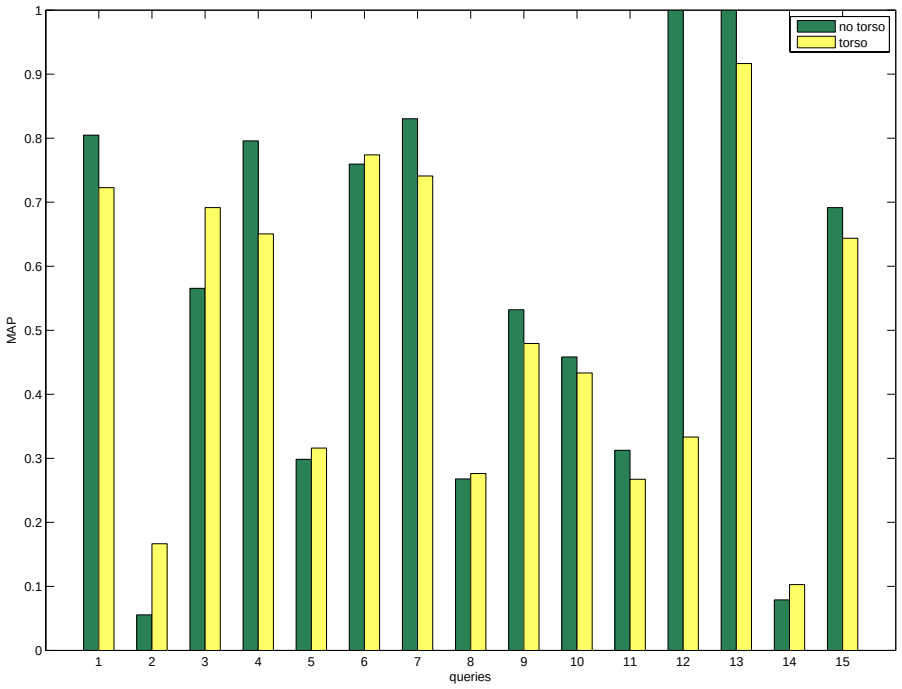



Figure 6.6: Mean Average Precision values of our method with respect to torso inclusion. The MAP of our method over the whole body is 0.501 when we query with the torso, whereas it is 0.5636 when we query over the limbs only. For some queries, including torso information increases performance slightly, however, on the overall, we see that using torso information is not very informative.

### 6.3.2 Controls

We cannot fairly compare to HMM models because complex activities require large numbers of states (which cannot be learned directly from data) to obtain a reasonable search vocabulary. However, discriminative methods are rather good at classifying activities without explicit dynamical models, and it is by no means certain that dynamical models are necessary (see section 2.2.5 in the discussion of related work). Discriminative models regard changes in the temporal structure of an action as likely to be small, and so well covered by training data. For this reason, we choose to compare with discriminative methods. There are three possible strategies, and we compare to each. First, one could simply identify activities from image-time features (like, for example, the work of [12, 26, 86]). Second, one could try to identify activities from lifted data, using lifted data to train models. Finally, one could try to identify activities from lifted data, but training using motion capture data.

#### 6.3.2.1 Control 1: SVM classifier over 2D tracks

To evaluate the effectiveness of our approach, we implemented an SVM-based action classifier over the raw 2D tracks. Using the tracker outputs for 17 videos as training set (chosen such that 2 different video sequences are available for each action), we built action SVMs for each limb separately. We used RBF kernel and 7 frame snippets of tracks to build the classifiers for this setting has given the best results for this control. A grid search over parameter space of the SVM is done and best classifiers are selected using 10-fold cross-validation. The performance of these SVMs are then evaluated over the remaining 56 videos. Figure 6.7 and Figure 6.8 shows the results. MAP value over the sets of queries is 0.3970 with Control 1. Note that for some queries, SVMs are quite successful in marking relevant documents. However, on the overall, SVMs are penalized by the noise and variance in dynamics of the activities. Our HMM limb activity models, on the other hand, deal with this issue by the help of the dynamics introduced by synthesized motion capture data. SVMs would need a great deal of training data to discover such dynamics. For some queries, SVM performances are good, however, on the overall, their precision and recall rate is low. Also, note that the

relevant videos are all scattered through the retrieval list.

#### 6.3.2.2 Control 2: SVM classifier over 3D lifts

We have also trained SVM classifiers over 3D lifted track points. Mean average precision of the whole query set in this case is 0.3963. This is not surprising, since there is some noise introduced by lifting 2d tracks, causing the performance of the classifier to be low. In addition, HMM method still has the advantage of using the dynamics introduced by motion capture dataset. The corresponding results are presented in Figure 6.7 and Figure 6.8. These results support the fact that motion capture dataset dynamics is a good clue for human action detection in our case.

#### 6.3.2.3 Control 3: SVM classifier over 3D motion capture set

Our third control is based on SVM classifiers built over 3D motion capture data set. We used the same vector-quantization as in building our HMM models, for generalization purposes. Mean average precision of the query set here is 0.3538. Although they rely on extra information added with the presence of motion capture data set, we observed that, these SVMs are also insufficient in tolerating the different levels of sustainability and different speeds of activities. This also causes problems with the composition. Generative nature of HMMs eliminate such difficulties and handle with varying length actions/activities easily.

## 6.4 Viewpoint evaluation

To evaluate our method's invariance to viewpoint, we queried 5 single activities (`jog, jump, jumpjack, reach, wave`) over the separate viewpoint data set that has 8 different view directions of subjects (Figure 6.2). We assume that if these simple sequences produce reliable results, the complex sequences will be accurate as well. Results of this evaluation are shown in Figures 6.9 and 6.10. As it can be seen, tracker

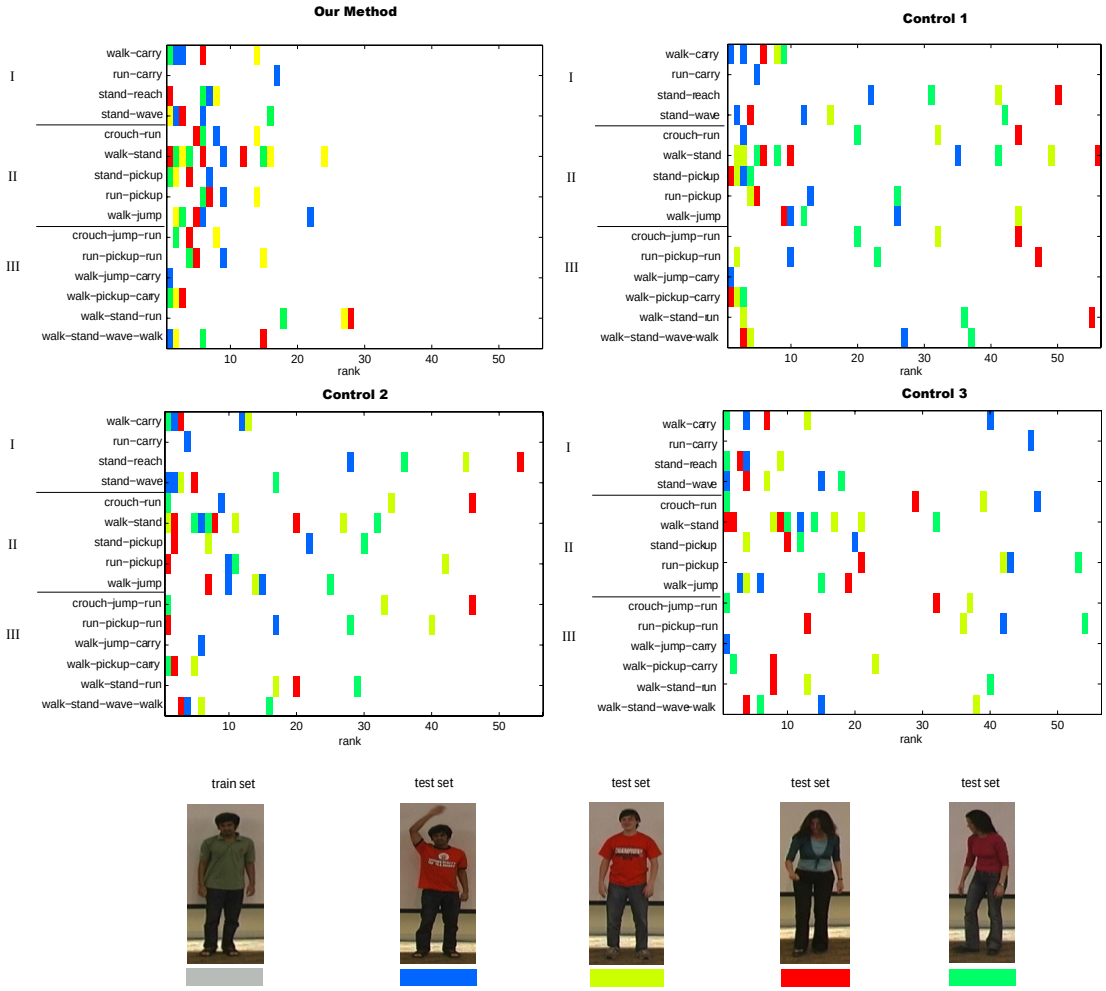



Figure 6.7: The results of ranking for 15 queries over our video collection. Our representation can give quite accurate results for complex activity queries, regardless of the clothing worn by the subject. In these images, a colored pixel indicates a relevant video. An ideal search would result in an image where all the colored pixels are on the left of the image. Each color represents a different outfit. We have three types of query here (see text for details). **Top left:** The ranking results of our activity modeling based on joint HMMs and motion capture data. Note that the videos retrieved in top columns are more likely to be relevant and the retrieval results are more condensed to the left. Note that the choice of the outfit doesn't affect the performance. **Top right:** Control 1: Separate SVM classifiers for each action over the 2D tracks of the videos. Composite queries built on top of a discriminative (SVM) based representation are not as successful as querying with our representation. Again, clothing does not affect the result. **Bottom left:** Control 2: SVM classifiers over 3D lifted tracks. **Bottom right:** Control 3: SVM classifiers over 3D motion capture data. While these classifiers benefit from dynamics of mocap data, they suffer due to lack of composition.

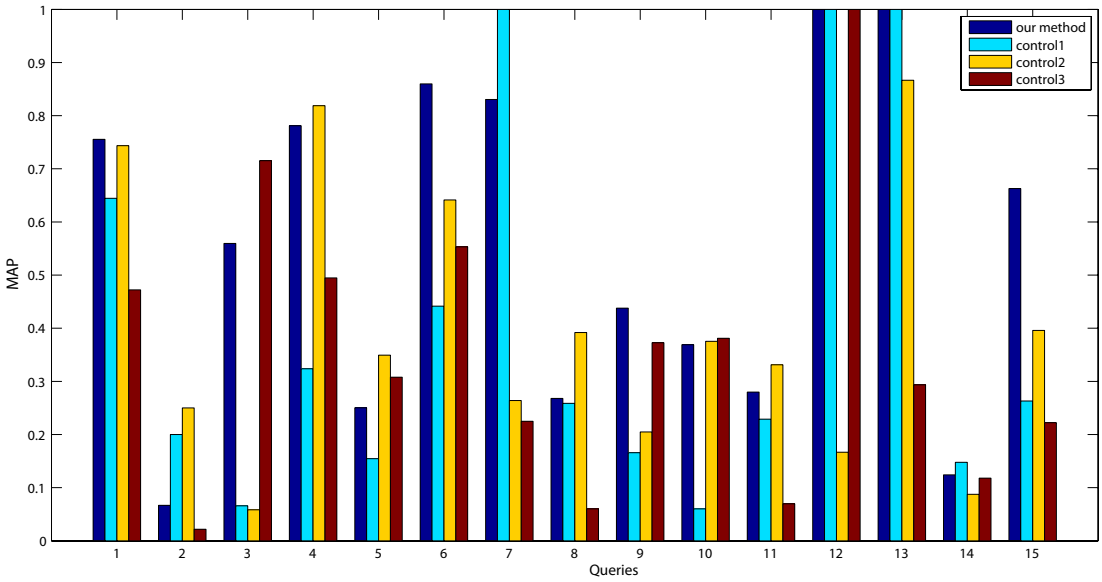



Figure 6.8: Average precision values for each query. Our method gives a mean average precision (MAP) of 0.5636 over the whole query set. Control 1's MAP value is 0.3970. Control 2 acquires a MAP of 0.3963, while it is 0.3538 for Control 3.

sometimes misses the moving arm, causing the performance of the system to degrade. However, we can say that on the overall, performance is not significantly affected with the change in viewpoint.

As Figure 6.9 shows, the performance is not significantly affected by the change in viewpoint, however there is slight lost of precision in some angles due to tracking and lifting difficulties in those view directions. Examples of non-reliable tracks are also shown in Figure 6.9. Due to occlusions and motion blur, the tracker tends to miss the moving arms quite often, making it hard to discriminate between actions.

Figure 6.10 shows the overall precisions averaged w.r.t. angles for each action. Not surprisingly, most confusion occurs between `reach` and `wave` actions. If the tracker misses the arm during these actions, it is highly likely that the dynamics of these actions will not be recovered and those two actions will resemble each another. On the other hand, `jumpjack` action is a combination of wave and jump actions, which is also subject to high confusion. Here, note that SVMs would need to be retrained for each viewing direction, while our method does not.

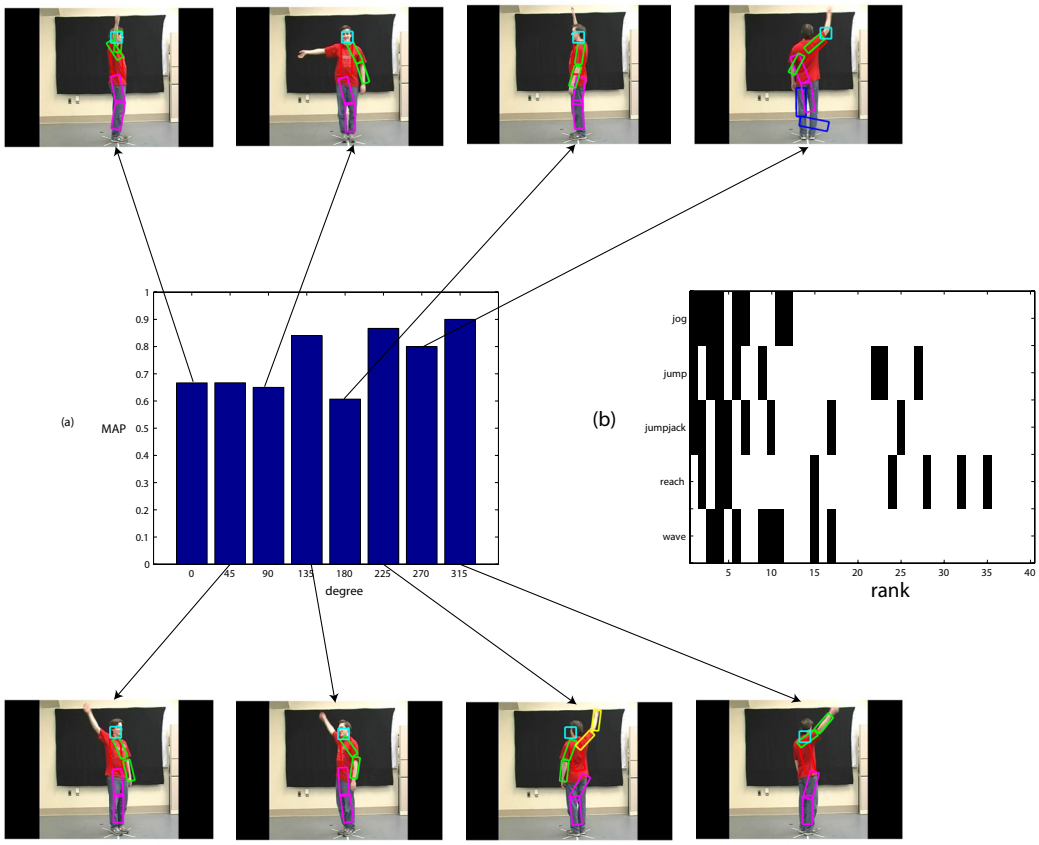
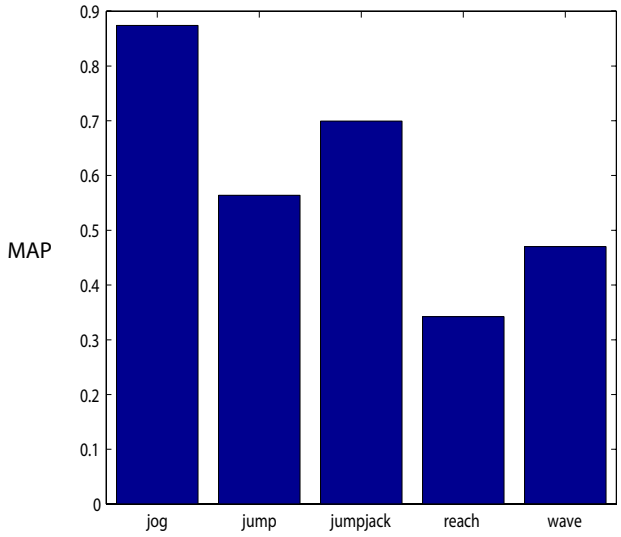
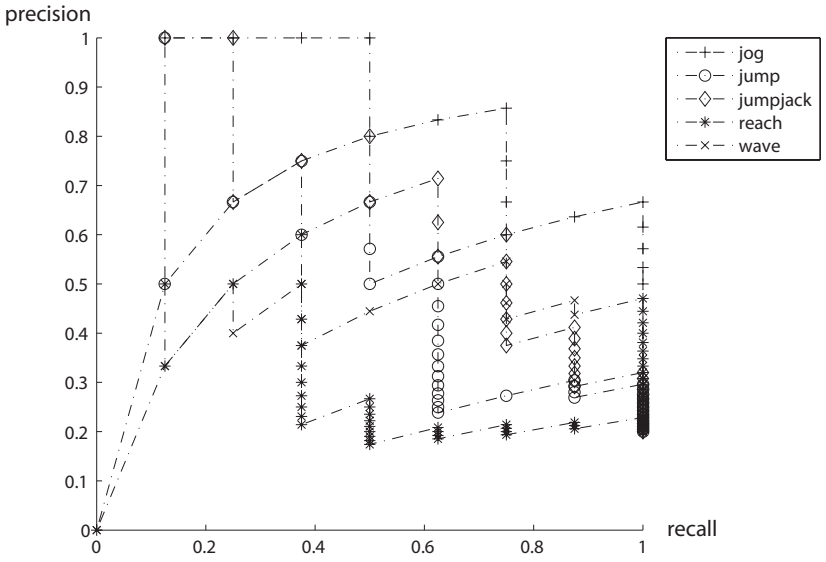



Figure 6.9: Evaluation of our method's sensitivity to viewpoint change. **(a)** Average precision values for each viewing direction. Some viewing directions has slightly worse performance due to the occlusion of the limbs and poor tracking response to bendings of the limbs in some view directions. Here, we show some representative frames with tracks for the `wave` action. **(b)** The ranking of the five queries of single actions separately. The poorest response comes from `reach` action, which inevitably confuses with `wave`, especially when the arms are out of track in the middle of the action. Here, note that SVMs would need to be retrained for each viewing direction, while our method does not.




(a)




(b)

Figure 6.10: (a) The mean precisions of each action averaged over the viewpoint change. The most confusion occurs between `reach` and `wave` actions. (b) Respective precision-recall curves for each action averaged over the angles. SVMs would need to be retrained for each viewing direction, while our method does not.

Figure 6.11: Example frames from the Friends dataset with relatively good tracks (which are superimposed).

## 6.5 Activity Retrieval with Complex Backgrounds

In order to see how well our algorithm will behave in football sequences with complicated settings, we tested our approach over football sequences taken from Friends TV Show. We have constructed a dataset, consisting of 19 short clips, in which characters play football in park (from Episode 9 of Season 3). We then annotated the actions of a single person in these clips by our available set of actions. This dataset is extremely challenging; the characters change orientation frequently, the camera moves, there are zoom-in and zoom-out effects and a complex and changing background. Examples frames from these sequences are shown in Fig. 6.11.

Since we built our activity models using a dataset of motion captured American football movements, we expect to have a higher accuracy in domains with similar actions. We test our system using 10 queries, ranging from simple to complex, and results are given in Fig. 6.12. For 9 out of 10 queries, the top retrieved video is a relevant video including the queried activity. Our MAP of 0.8172 over this dataset shows that our system is quite good in retrieving football movements, even in complicated settings.

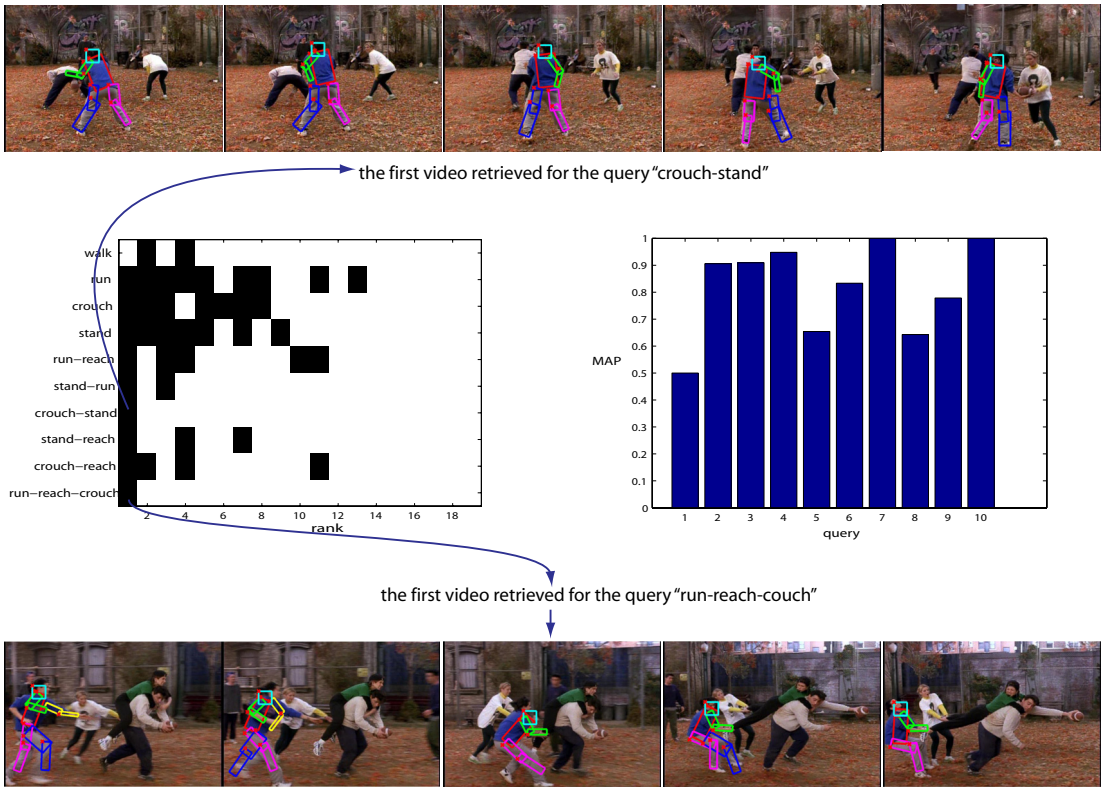



Figure 6.12: Results of our retrieval system over the Friends dataset. Our system is quite successful over this dataset. Since our activity models are formed using motion capture dataset which consists of American football movements, this dataset is a natural application domain for our system. In 9 out of 10 queries, our system returns a relevant video as the top result and we achieve a MAP of 0.8172 over this dataset.

# Chapter 7

# Conclusions and Discussion

We believe that for simple actions, when there is no composition or viewpoint change, we do not need complex models of dynamics or dense templates for matching. We show with empirical evidence that compact representations may suffice, plus, may outperform intricate models both in terms of precision and runtime.

Within this framework, we describe how one can make use of the rectangular region information together with simple velocity features, in the presence of silhouettes. Our new pose-descriptor is based on the orientation of body parts; we extract the rectangular regions from a human silhouette and form a spatial oriented histogram of these rectangles. Our results are directly comparable and even superior to the results presented over the state-of-art action datasets. When pose itself is not enough for discriminating between actions, we have shown how to boost the performance by including simple velocity features and build a hierarchical model on top of our classification scheme. We have demonstrated how we can obtain efficient action recognition with the minimum of dynamics. Result is an intuitive and fast action recognition system with high accuracy rates, even in challenging conditions like camera zoom, camera movement, lighting conditions and different outfits.

Our experiments on single action recognition have shown that the human pose encapsulates many useful pieces of information about the action itself. Following this observation, we took one step further and evaluate this claim within the still images.

Our results are promising, and we have illustrated that as the humans can perceive actions from looking at a single photograph, machines can also do so, to a certain extent.

When the silhouettes are not easily extractable or very noisy, one alternative would be to use the boundary-fitted lines, instead of rectangular regions. We make use of orientation statistics of such lines and a compact representation of optical flow for single action recognition. In our experiments, we have observed that, short lines describe the fine details of the pose of the body, whereas longer lines give some clue about the coarser shape. Our shape features use a concatenation of these two forms for better recognition. We also observed that shape and motion cues are complimentary to each other, and we can further enhance the performance by treating them together in a discriminative manner. We also observed that dense templates for optical flow is not much needed (as used in [26, 49]) and we can get quite enough information by using spatial and directional binning of the flow. This usage, together with maximum entropy selection of features, densifies the feature dimensionality a great deal, thus, reducing the classification time. Most of the existing algorithms suffer shortcomings of processing with large feature dimensions, since they have pixel-wise or very dense templates. This feature reduction is, therefore, quite useful; it opens up room for application of more sophisticated classification schemes.

There is little evidence that a fixed taxonomy for human motion is available. However, research to date has focused on multi-class discrimination of simple actions. Everyday activities are more complex in nature and people tend to perform composite activities both on the spatio and temporal dimensions. In our second scenario, we handled this aspect of human activity understanding. We have demonstrated a representation of human motion that can be used to query for complex activities in a large collection of video. We build our queries using finite state automata and for each limb, we write separate queries. We are aware of no other method that can give comparable responses for such queries.

Our representation uses a generative model, built using motion capture and applying it over video data. This can also be thought as an instance of transfer learning; we

transfer the knowledge we gain from 3D motion capture data, to 2D everyday activity data. This transfer learning helps a lot for building composite activity models; if we did not use transfer learning, we would need a considerable amount of videos for training body part models, which is hard to acquire. By this way, we can use any set of motion capture data and broaden our set of activities easily. Furthermore, by joining models of atomic actions to form activity models, we do not need the train examples for complex activity sequences and we perform minimum parameter estimation.

One of the strengths of our method is that, when searching for a particular activity, no example activity is required to formulate a query. We use a simple and effective query language; we simply search for activities by formulating sentences like "Find action $X$ followed by action $Y$" or "Find videos where legs doing action $X$ and arms doing action $Y$" via finite state automata. Matches to the query are evaluated and ranked by the posterior probability of a state representation summed over strings matching the query. Using a strategy like ours, one can search for activities that have never been seen before.

As our results show, query responses are unaffected by clothing, and our representation is robust to aspect. Our representation significantly outperforms discriminative representations built using image data alone. It also outperforms models built on 3D lifted responses, meaning that the dynamics transferred from motion capture domain to real world domain helps in retrieval of complex activities. In addition, the generative nature of HMM models helps to compansate the different levels of sustainability of the actions and makes composition across time easier.

Moreover, since our representation is in 3D, we don't need to retrain our models separately for each viewing direction. We show that our representation is mostly invariant to change in viewing direction.

## 7.1 Future Directions

This work introduces our baby steps for understanding of the human actions/activities. This highly applicable topic is still at its infancy and much is left to be done. However,

we think that we have delineated important directions for both simple and complex activity recognition.

This work is extensible in many ways, such as:

- Rectangle-based pose description can be augmented to handle the view-invariance case, by means of tracking rectangular regions in temporal dimension and using orthographic projections of them. The volumetric details of the descriptor can be investigated this way.
- While forming our complex activity models, the biggest difficulty we faced was to properly track the fast moving limbs and then lifting to 3D in the presence of such tracking errors and ambiguities. That's why we can say that there is much room for improvement; a better tracker would give better results immediately. We think that this improved tracker should be appearance-based full body tracker, whereas it might be enhanced by the use of a coarse optical flow model of the scene.
- After tracking, the second source of noise comes from lifting 2D tracks to 3D. An improved and unambiguous lifting mechanisms can help to improve the performance of our system.
- More discriminative features can be used as a front-end to our complex activity recognition system (after [91, 90]). Currently, we use only 3D joint points in a loosely generative way. Addition of more features is very likely to uplift the descriptive performance of the system.
- Our current vocabulary for defining activities are limited to that of the motion capture data. Enriching this vocabulary would make the system more comprehensible and qualified for application to everyday activities. In addition, current activity recognition research lacks the presence of a suitable vocabulary and ontology for actions. We would definitely benefit from some theory about how a canonical action vocabulary could be built, and how an accurate ontology – which describes the general principles of the mutual relationships of actions [21]– can be formed.

- Currently, we deal with the activity path of a single subject inside the videos. However, real-world data may involve more than one person. In a similar fashion as we do, multiple subjects can be tracked and modelled for semantic analysis of their activity pathways. Furthermore, modelling interactions between people(as in [68, 39, 40]) would extend the capabilities of the our complex activity system. In this way, a higher level understanding of scene and activity semantics would be possible.

Research in this thesis can be applied to many domains. There are recently developing systems(such as [107] which has initiated as an multi-sensor system for detecting violence in videos) that can benefit from modeling body parts in the way that we do. Additionally, video hosting websites like YouTube would improve their searching capabilities with a searching framework similar to ours. Since our search mechanism need no visual examples, but pure organized text, our query system is directly adoptable to their structure.

## 7.2 Relevant Publications

- Nazlı İkizler and David A. Forsyth, "Searching for Complex Human Activities with No Visual Examples, accepted for publication in *International Journal of Computer Vision (IJCV)*, 2008.
- Nazlı İkizler and Pınar Duygulu, "Histogram of Oriented Rectangles: A New Pose Descriptor for Human Action Recognition", submitted to *Journal of Image and Vision Computing(IMAVIS)*.
- Nazlı İkizler and Pınar Duygulu, "Human action recognition using distribution of oriented rectangular patches", *2nd Workshop on Human Motion Understanding, Modeling, Capture and Animation In Conjunction with Eleventh IEEE International Conference on Computer Vision (ICCV 2007)*, October 2007.
- Nazlı İkizler and David A. Forsyth, "Searching video for complex activities with finite state models", In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

- Nazlı İkizler, R. Gökberk Cinbiş and Pınar Duygulu, “Action Recognition with Line and Flow Histograms”, submitted.
- Nazlı İkizler, R. Gökberk Cinbiş, Selen Pehlivan and Pınar Duygulu, “Recognizing Actions from Still Images”, submitted.

# Bibliography


[1] Youtube. http://www.youtube.com.

[2] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[3] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *PAMI*, 28(1), January 2006.

[4] J. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, March 1999.

[5] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

[6] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853, 2005.

[7] O. Arikan, D. Forsyth, and J. O'Brien. Motion synthesis from annotations. In *Proc of SIGGRAPH*, 2003.

[8] O. Arikan and D. A. Forsyth. Interactive motion generation from examples. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 483–490. ACM Press, 2002.

[9] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard. Segmenting motion capture data into distinct behaviors. In *GI '04: Proceedings of the 2004 conference on Graphics interface*, pages 185–194, School of

Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

[10] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram. Human activity recognition using multidimensional indexing. *IEEE T. Pattern Analysis and Machine Intelligence*, 24(8):1091–1104, August 2002.

[11] A. Bissacco, M.-H. Yang, and S. Soatto. Detecting humans via their pose. In *Proc. Neural Information Processing Systems*, 2006.

[12] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Int. Conf. on Computer Vision*, pages 1395–1402, 2005.

[13] A. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. *Proc. Roy. Soc. B*, 352:1257–1265, 1997.

[14] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE T. Pattern Analysis and Machine Intelligence*, 23(3):257–267, March 2001.

[15] A. Bobick and A. Wilson. A state based technique for the summarization and recognition of gesture. In *Int. Conf. on Computer Vision*, pages 382–388, 1995.

[16] A. Bobick and A. Wilson. A state based approach to the representation and recognition of gesture. *IEEE T. Pattern Analysis and Machine Intelligence*, 19(12):1325–1337, December 1997.

[17] A. F. Bobick and Y. A. Ivanov. Action recognition using probabilistic parsing. In *CVPR*, page 196, 1998.

[18] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE T. Pattern Analysis and Machine Intelligence*, 22(8):844–851, August 2000.

[19] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[20] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[21] R. Chellappa, A. Roy Chowdhury, and S. Zhou. Recognition of humans and their activities using video. In *Morgan Claypool*, 2005.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages I: 886–893, 2005.

[23] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*, Graz, Austria, May 2006.

[24] Y. Dedeoglu, B. Toreyin, U. Gudukbay, and A. E. Cetin. Silhouette-based method for object classification and human action recognition in video. In *International Workshop on Human-Computer Interaction(CVHCI06) held in conjunction with ECCV, LNCS 3979*, pages 64–77, Graz, Austria, 2006.

[25] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005.

[26] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV '03*, pages 726–733, 2003.

[27] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *17th SIGKDD Conf. on Knowledge Discovery and Data Mining*, 2004.

[28] A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2007.

[29] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *Int. J. Computer Vision*, 61(1):55–79, January 2005.

[30] X. Feng and P. Perona. Human action recognition by sequence of movelet codewords. In *3D Data Processing Visualization and Transmission*, pages 717–721, 2002.

[31] A. Fod, M. J. Matarić, and O. C. Jenkins. Automated derivation of primitives for movement classification. *Auton. Robots*, 12(1):39–54, 2002.

[32] D. Forsyth and M. Fleck. Body plans. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 678–683, 1997.

[33] D. Forsyth, O.Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan. Computational studies of human motion i: Tracking and animation. *Foundations and Trends in Computer Graphics and Vision*, 1(2/3):1–255, 2006.

[34] D. Forsyth and J. Ponce. *Computer Vision: a modern approach*. Prentice-Hall, 2002.

[35] W. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition*, June 1995.

[36] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999.

[37] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE T. Pattern Analysis and Machine Intelligence*, 22:809–830, 2000.

[38] P. Hong, M. Turk, and T. Huang. Gesture modeling and recognition using finite state machines. In *Int. Conf. Automatic Face and Gesture Recognition*, pages 410–415, 2000.

[39] S. Hongeng and R. Nevatia. Multi-agent event recognition. *Int. Conf. on Computer Vision*, 02:84, 2001.

[40] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129–162, November 2004.

[41] N. Howe. Silhouette lookup for automatic pose tracking. In *IEEE Workshop on Articulated and Non-Rigid Motion*, page 15, 2004.

[42] N. R. Howe, M. E. Leventon, and W. T. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *Proc. Neural Information Processing Systems*, pages 820–26, 2000.

[43] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE transactions on systems, man, and cyberneticspart c: applications and reviews*, 34(3), 2004.

[44] L. Ikemoto, O. Arikan, and D. Forsyth. Quick transitions with cached multi-way blends. In *ACM Symposium on Interactive 3D Graphics and Games (I3D)*, 2007.

[45] N. Ikizler and P. Duygulu. Human action recognition using distribution of oriented rectangular patches. In *Human Motion Workshop (held in conjunction with ICCV) LNCS 4814*, pages 271–284, 2007.

[46] S. Ioffe and D. Forsyth. Learning to find pictures of people. In *Proc. Neural Information Processing Systems*, 1998.

[47] O. C. Jenkins and M. J. Matarić. Automated derivation of behavior vocabularies for autonomous humanoid motion. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 225–232, New York, NY, USA, 2003. ACM Press.

[48] O. C. Jenkins and M. J. Matarić. A spatio-temporal extension to isomap nonlinear dimension reduction. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 56, New York, NY, USA, 2004. ACM Press.

[49] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *Int. Conf. on Computer Vision*, 2007.

[50] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *Int. Conf. on Computer Vision*, 2007.

[51] Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal shape and flow correlation for action recognition. In *Visual Surveillance Workshop*, 2007.

[52] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 473–482. ACM Press, 2002.

[53] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, page 432, Washington, DC, USA, 2003. IEEE Computer Society.

[54] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data. In *Proc of SIGGRAPH*, 2002.

[55] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Computer Vision*, 43(1):29–44, 2001.

[56] Y. Li, T. Wang, and H.-Y. Shum. Motion texture: a two-level statistical model for character motion synthesis. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 465–472. ACM Press, 2002.

[57] H. Ling and K. Okada. Diffusion distance for histogram comparison. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 246–253, 2006.

[58] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI*, 26, 2004.

[59] M. J. Matarić, V. B. Zordan, and Z. Mason. Movement control methods for complex, dynamically simulated agents: Adonis dances the macarena. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, pages 317–324, New York, NY, USA, 1998. ACM Press.

[60] M. J. Matarić, V. B. Zordan, and M. M. Williamson. Making complex articulated agents dance. *Autonomous Agents and Multi-Agent Systems*, 2(1):23–43, 1999.

[61] D. J. Moore. Vision-based recognition of actions using context. Technical report, Georgia Institute of Technology, 2000. PhD Thesis.

[62] T. Mori, Y. Segawa, M. Shimosaka, and T. Sato. Hierarchical recognition of daily human actions based on continuous hidden markov models. In *Int. Conf. Automatic Face and Gesture Recognition*, pages 779–784, 2004.

[63] E. Muybridge. Animal locomotion, 1887.

[64] E. Muybridge. *The Human Figure in Motion*. Dover, 1989.

[65] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[66] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *British Machine Vision Conference*, 2006.

[67] N. Oliver, A. Garg, and E. Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding*, 96(2):163–180, November 2004.

[68] N. Oliver, B. Rosario, and A. Pentland. Graphical models for recognizing human interactions. In *Proc. Neural Information Processing Systems*, Denver, Colorado, USA, November 1998.

[69] C. Pinhanez and A. Bobick. Pnf propagation and the detection of actions described by temporal intervals. In *DARPA IU Workshop*, pages 227–234, 1997.

[70] C. Pinhanez and A. Bobick. Human action detection using pnf propagation of temporal constraints. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 898–904, 1998.

[71] R. Polana and R. Nelson. Detecting activities. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2–7, 1993.

[72] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[73] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–286, February 1989.

[74] D. Ramanan. Learning to parse images of articulated bodies. In *Proc. Neural Information Processing Systems*, 2006.

[75] D. Ramanan and D. Forsyth. Automatic annotation of everyday movements. In *Proc. Neural Information Processing Systems*, 2003.

[76] D. Ramanan, D. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages I: 271–278, 2005.

[77] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE T. Pattern Analysis and Machine Intelligence*, 29(1):65–81, 2007.

[78] L. Ren, A. Patrick, A. A. Efros, J. K. Hodgins, and J. M. Rehg. A data-driven approach to quantifying natural human motion. *ACM Trans. Graph.*, 24(3):1090–1097, 2005.

[79] X. Ren, A. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. *Proc. ICCV*, pages 824–831, 2005.

[80] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *European Conference on Computer Vision*, page IV: 700 ff., 2002.

[81] R. Rosales and S. Sclaroff. Specialized mappings and the estimation of body pose from a single image. In *IEEE Human Motion Workshop*, pages 19–24, 2000.

[82] C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph. Appl.*, 18(5):32–40, 1998.

[83] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Computer Vision*, 40(2):99–121, 2000.

[84] M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2007.

[85] A. Samal and P. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25(1):65–77, 1992.

[86] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society.

[87] G. Shakhnarovich and P. V. T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Int. Conf. on Computer Vision*, 2003.

[88] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation networks for recognition of partially ordered sequential action. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[89] J. M. Siskind. Reconstructing force-dynamic models from video sequences. *Artificial Intelligence*, 151:91–154, 2003.

[90] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional random fields for contextual human motion recognition. In *ICCV*, pages 1808–1815, 2005.

[91] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. *IEEE Conf. on Computer Vision and Pattern Recognition*, 1:390–397, 2005.

[92] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE T. Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[93] C. Thurau. Behavior histograms for action recognition and human detection. In *Human Motion Workshop LNCS 4814*, pages 299–312, 2007.

[94] B. Toreyin, Y. Dedeoglu, and A. E. Cetin. Hmm based falling person detection using both audio and video. In *IEEE International Workshop on Human-Computer Interaction(CVHCI05) held in conjunction with ICCV, LNCS 3766*, pages 211–220, Beijing, China, 2005.

[95] D. Tran and D. Forsyth. Configuration estimates improve pedestrian finding. In *Proc. Neural Information Processing Systems*, 2007.

[96] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[97] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[98] D. D. Vecchio, R. Murray, and P. Perona. Decomposition of human motion into dynamics-based primitives with application to drawing tasks. *Automatica*, 39(12):2085–2098, 2003.

[99] L. Wang and D. Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2007.

[100] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[101] A. Wilson and A. Bobick. Learning visual behavior for gesture analysis. In *IEEE Symposium on Computer Vision*, pages 229–234, 1995.

[102] A. Wilson and A. Bobick. Parametric hidden markov models for gesture recognition. *IEEE T. Pattern Analysis and Machine Intelligence*, 21(9):884–900, September 1999.

[103] S.-F. Wong, T.-K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2007.

[104] J. Yamato, J. Ohya, and K. Ishii. Recognising human action in time sequential images using hidden markov model. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[105] W. Yan and D. Forsyth. Learning the behavior of users in a public space through video tracking. In *WACV05*, pages I: 370–377, 2005.

[106] J. Yang, Y. Xu, and C. S. Chen. Human action learning via hidden markov model. *IEEE Transactions on Systems Man and Cybernetics*, 27:34–44, 1997.

[107] W. Zajdel, D. Krijnders, T. Andringa, and D. Gavrila. Cassandra: Audio-video sensor fusion for aggression detection. In *IEEE Int. Conf. on Advanced Video and Signal based Surveillance (AVSS)*, London, 2007.

[108] J. Zhang, J. Luo, R. Collins, and Y. Liu. Body localization in still images using hierarchical models and hybrid search. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[109] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE T. Pattern Analysis and Machine Intelligence*, 26(9):1208–1221, September 2004.

[110] W.-Y. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computer Surveys*, 35(4):399–458, 2003.

[111] G. Zhu, C. Xu, Q. Huang, and W. Gao. Action recognition in broadcast tennis video. In *Proceedings IAPR International Conference on Pattern Recognition*, 2006.

[112] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. *IEEE Conf. on Computer Vision and Pattern Recognition*, 1(2):4, 2006.